



HAL
open science

Design under constraints of Dependability and Energy for Wireless Sensor Network

van Trinh Hoang

► **To cite this version:**

van Trinh Hoang. Design under constraints of Dependability and Energy for Wireless Sensor Network. Electronics. Université de Bretagne Sud, 2014. English. NNT : 2014LORIS351 . tel-01193330

HAL Id: tel-01193330

<https://theses.hal.science/tel-01193330>

Submitted on 4 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE / UNIVERSITÉ DE BRETAGNE SUD

sous le sceau de l'Université européenne de Bretagne

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE SUD

Mention : Electronique

École Doctorale SICMA

présentée par

Van-Trinh HOANG

Préparée au Laboratoire Lab-STICC
Lorient. France

Design under constraints of Dependability and Energy for Wireless Sensor Network

Thèse soutenue le 08 Décembre 2014

devant le jury composé de :

Yvon TRINQUET

Professeur, IRCCyN - Ecole Centrale de Nantes

Sébastien PILLEMENT

Professeur, Université de Nantes/ *Rapporteur*

Michel COMBACAU

Professeur, LAAS-CNRS/ *Rapporteur*

Eric SENN

Maître de Conférences/HDR, Université de Bretagne Sud

Dominique JUTEL

Responsable d'innovation, société ERYMA

Nathalie JULIEN

Professeur, Université de Bretagne Sud/ *Directrice de thèse*

Pascal BERRUET

Professeur, Université de Bretagne Sud/ *Co-directeur de thèse*



LAB-STICC, UNIVERSITÉ DE BRETAGNE-SUD
ECOLE DOCTORALE : SICMA

THESE

pour obtenir le titre de

DOCTEUR EN SCIENCES

de l'université de Bretagne-Sud - Lorient

MENTION : ELECTRONIQUE

Présentée et soutenue par

Van-Trinh HOANG

Design under constraints of Dependability and Energy for Wireless Sensor Network

Thèse dirigée par Pr. Nathalie JULIEN et Pr. Pascal BERRUET

Laboratoire Lab-STICC, Université de Bretagne-Sud - Lorient

Soutenue en 08 Décembre 2014

Jury:

Yvon TRINQUET	Professeur, IRCCyN - Ecole Centrale de Nantes
Sébastien PILLEMENT	Professeur, Université de Nantes/ Rapporteur
Michel COMBACAU	Professeur, LAAS-CNRS/ Rapporteur
Eric SENN	MCF/HDR, Université de Bretagne-Sud
Dominique JUTEL	Responsable d'innovation, société ERYMA
Nathalie JULIEN	Professeur, Université de Bretagne Sud/ Directrice de thèse
Pascal BERRUET	Professeur, Université de Bretagne Sud/ Co-directeur de thèse

ACKNOWLEDGEMENTS

This thesis has been written at Lab-STICC research center at university of South-Brittany in Lorient, France, during the spring semester 2014. One of the joys of completion is to look over the journey past and remember all the frequent helps from my colleagues of the Department of Electronic Engineering at Lab-STICC laboratory and of university of South-Brittany. I am thankful to the Brittany region and General Council of Morbian (CG56) with the support of ERYMA Company for providing funding throughout my PhD career.

I would like to express my special gratitude towards my thesis director, Professor Nathalie JULIEN, for being a constant source of inspiration, invaluable encouragement and aid during my three years at Lab-STICC. I am sincerely thankful to my co-supervisor, Professor Pascal BERRUET who has guided me through this thesis work, patiently answered my many persistent questions and provided insightful discussions about the research. I am very grateful for scientific advices, insightful knowledge and suggestions from my supervisors. The mentoring, friendship and collegiality of my both two Professors enriched my academic life and have left a profound impression on how academic research and collaboration with industrial partner should ideally be conducted.

I would also like to thank Professor Michel COMBACAU and Professor Sébastien PILLEMENT for being my reviewers and thoroughly reading and acknowledging my thesis. In addition, I would like to thank Professor Yvon TRINQUET and Dr.Sc. Eric SENN for being jury. I would like to express my sincere gratitude to our industrial partner, Dominique JUTEL - innovation manager of ERYMA Company for his precious scientific advices and discussions during our annual meetings. Thanks to his help, I have had inspiring ideas to move forward during implementation experimentations of my thesis work. Reviewing a thesis work is not easy task and I am grateful for their detailed reviewing.

Lab-STICC has provided a rich and fertile environment to study and explore new ideas. At Lab-STICC, I would like to thank Florent de LAMOTTE who has been extremely supportive in allowing me to participate in university teaching activities during my three years of thesis. These precious experiences have taught me how to transfer information to others. I am grateful to Philippe GIQUEL, Valère ALIBERT and other colleagues for their meaningful helps and discussion during my thesis work. Sincerely, I am proud to be part of the lab and thank for welcoming me as a friend and helping to develop the ideas in this thesis.

Finally, I would like to express my special gratitude for my beloved parents and brother for their precious encouragement that helped me to pass through the difficult moments.

ABSTRACT

The uncertain contexts in which recent WSN embedded applications evolve have big impact on these applications. Traditionally, the objective of availability generally doubles hardware and functional redundancy; it means that the overhead is doubled in term of energy and cost. Besides, wireless node system is powered by limited battery; hence power consumption parameter is only set to a number of components and functionalities at minimum resources. However, due to the technology reduction, process variability conducts to increase the possibility of failures. In order to guarantee an acceptable quality of service for the users, and on the operating lifetime of the system, it should carry studies at the upper phases involving both dependability and consumption constraints.

This thesis aims to propose novel design for wireless sensor networks, in order to reduce energy consumption and to increase network dependability. First of all, these challenges are considered at sensor node layer by integrating a Power and Availability Manager (PAM) device in node system. Based on Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS) policies, a method is proposed that allows the PAM to monitor power and energy, so as to extend the battery life and load requirements. In the other hand, we present self-diagnosis including hardware failure detection and localization for discrete event system as wireless node system. The detection of hardware failure is carried out by supervising the current measurement in the real-time operation. Then the failure localization is run by executing the functional tests. After hardware failure detection and localization steps, the PAM executes system reconfiguration to replace failed component by the complementary material, or to switch to other operating mode. The simulation and implementation results of this approach reveal many interesting and notable conclusions that can serve as prerequisites for energy-efficiency and dependability enhancement strategies.

In the second part, this work also presents the improved approach for cluster-head selection to deal with cluster-head unavailability when WSN is deployed using cluster topology. This selection algorithm takes both residual energy, distance to the base station and obstacle-awareness into account, which allows selecting the most appropriate candidate nodes to become cluster-heads. The simulation results indicate significant improvement in network throughput, lifetime and load energy balance between sensor nodes in the network. Finally, several global discussions and conclusions based on the rich results from this work are given.

Keywords : Energy-efficiency, dependability, wireless sensor network, reconfiguration, FPGA, discrete event systems.

RÉSUMÉ

Le contexte incertain dans lequel évoluent les applications embarquées influence fortement ces dernières. L'objectif de disponibilité induit généralement une forte redondance matérielle et fonctionnelle. A l'inverse, le paramètre de consommation prône un nombre et un fonctionnement à minima des ressources. Avec la réduction de la technologie, la variabilité des procédés de fabrication induit la possibilité accrue de défaillances. De façon à garantir une qualité de service acceptable par l'utilisateur, et ce sur la totalité de la durée de vie du circuit, il convient de mener des études associant dès les phases amont les deux paramètres sûreté de fonctionnement et consommation.

Cette thèse a pour objectif de proposer une nouvelle conception pour les réseaux de capteurs sans fil, afin de réduire la consommation d'énergie et d'augmenter la fiabilité du réseau. Tout d'abord, ces défis sont considérés à la couche de noeud de capteur en intégrant un dispositif appelé Power and Availability Manager (PAM) dans le système de noeud. Basé sur les techniques de Dynamic Power Management (DPM) et de Dynamic Voltage and Frequency Scaling (DVFS), on propose une méthode qui permet au PAM de contrôler l'énergie consommée effectivement, afin d'étendre la durée de vie de la batterie et les exigences de charge. De l'autre côté, nous présentons l'autodiagnostic, y compris la détection et la localisation de défaillance matérielle pour le noeud de capteur considéré comme un système à événements discrets. La détection d'une panne matérielle est réalisée en surveillant la mesure du courant en temps réel. Ensuite, la localisation de la panne est gérée par l'exécution des tests fonctionnels. Après avoir détecté et localisé une panne matérielle, le PAM exécute la reconfiguration du système pour remplacer le composant défaillant par le matériel complémentaire, ou pour passer à un autre mode de fonctionnement. Les résultats de la simulation et de la mise en oeuvre de cette approche révèlent des conclusions intéressantes et remarquables qui peuvent servir comme les conditions préalables aux stratégies d'efficacité énergétique et d'amélioration de la fiabilité.

Dans la deuxième partie, ce travail présente l'approche pour la sélection du cluster-head pour résoudre le problème de l'indisponibilité du cluster-head quand le WSN est déployé à l'aide de la topologie de cluster. Cet algorithme de sélection prend en compte à la fois les termes de l'énergie résiduelle, de la distance vers la station de base et de la sensibilisation d'obstacle, ce qui permet de sélectionner les noeuds les plus appropriés pour devenir cluster-heads. Les résultats de simulation montrent une amélioration significative au débit du réseau, à la durée de vie et à la distribution équilibrée de l'énergie entre les noeuds dans le réseau. Enfin, plusieurs discussions et des conclusions basées sur les résultats riches de ce travail sont données.

Mots clés : Efficacité énergétique, sûreté de fonctionnement, réseau de capteurs sans fil, reconfiguration, FPGA, systèmes à événements discrets.

Contents

1	INTRODUCTION	11
1.1	Wireless Sensor Network	12
1.2	Trade Market	14
1.3	Study Objective	15
1.4	Structure of the Thesis Dissertation	17
2	RESEARCH CONTEXT	19
2.1	Network Assumptions	19
2.1.1	Communication topologies	19
2.1.2	WSN characteristics	21
2.2	Problem Statement	21
2.3	Reference Works	23
2.3.1	Energy-efficiency	23
2.3.2	Availability	24
2.3.3	Conclusion	29
2.4	Novel Contributions	29
3	BASIS BACKGROUND	31
3.1	Constraints and Impacts	31
3.1.1	Ineffective use of energy	31
3.1.2	Hardware failures	31
3.2	Methods for Power and Energy Management	32
3.2.1	Sources of energy dissipation	32
3.2.2	Dynamic power management	34
3.2.3	Dynamic voltage and frequency scaling	34
3.3	Reconfiguration	38
3.3.1	Detection and localization	38
3.3.2	Context analysis	39
3.3.3	Decision making	40
3.3.4	Decision implementation	40
3.4	Modeling Tools of Discrete Event Systems (DES)	40
3.4.1	DES introduction	40

3.4.2	Tools for DES modeling	43
3.5	Conclusion	49
4	APPROACH AT NODE LEVEL	51
4.1	Introduction	52
4.1.1	Novel hardware design of wireless sensor node	52
4.1.2	Issues and corrective solutions	53
4.2	Approach for Energy-Efficiency and Availability Improvement	54
4.2.1	Energy management	54
4.2.2	Availability management	55
4.2.3	Node functionality modeling	60
4.3	Simulation Experiment	62
4.3.1	Energy-saving	62
4.3.2	Availability improvement	70
4.4	Implementation Experiment	77
4.4.1	PAM material implementation	77
4.4.2	Adjustable regulator implementation	84
4.4.3	Hardware failure detection implementation	91
4.5	Conclusion	95
5	APPROACH AT NETWORK LEVEL	97
5.1	Cluster Structure Extension	98
5.2	Major Factors in Cluster-Head Selection Algorithm	99
5.2.1	Factor of residual energy	99
5.2.2	Factor of distance to Sink	100
5.2.3	Factor of obstacle-aware	101
5.3	Obstacle-Aware Cluster-Head Selection Approach	102
5.4	Approach for Sink Unavailability Issue	104
5.5	Evaluation	106
5.5.1	Simulation tool	106
5.5.2	Network simulation model	108
5.5.3	Simulation results	111
5.6	Conclusion	112
6	DISCUSSION AND CONCLUSION	114
6.1	Impact Factors on Energy and Availability for Wireless Sensor Node	114
6.1.1	Operating Points Impact on Energy Reduction	115
6.1.2	Switching Latencies of State Transitions	115
6.1.3	Measurement Sensitivity Impact on Hardware Failure Detection	116
6.1.4	Component Endurance Impact on Node System	117
6.2	Impact Factors on Energy and Availability for Wireless Sensor Network	118
6.3	Conclusion and Perspectives	122
6.3.1	Conclusion	122

6.3.2 Perspectives 125
6.4 List of Publications 127

List of Figures

1.1	Hardware configuration of wireless sensor node	12
1.2	Processing steps of data acquisition and actuation	13
1.3	WSN applications in daily life	15
1.4	Revenu and growth rate of WSN trade market	16
2.1	Structure of cluster topology	20
2.2	Diagram of CSMA-CA communication protocol	22
2.3	Taxonomy framework for fault detection techniques	26
2.4	Mesh topology for network deployment	28
2.5	Classification of energy and availability approaches for WSN	30
3.1	Switching power dissipation in an inverter circuit	34
3.2	Task graph mapping and scheduling example	35
3.3	Trade-off between supply voltage and charging circuit delay	36
3.4	Example of using DVFS in the scheduling process	37
3.5	Reconfiguration process diagram	39
3.6	Warehouse system	42
3.7	Sample path of the warehouse system	43
3.8	Diagram of state transitions	45
3.9	Diagram of statechart	46
3.10	Marked Petri Net	48
3.11	Timed Stochastic Petri Net Modeling	50
4.1	Block diagram of wireless sensor node	52
4.2	Principle of DPM and DVFS implementation in wireless sensor node	55
4.3	Principle of first current measurement implementation	56
4.4	Real-time current measurement using HALL probe	57
4.5	Hardware failure localization process	59
4.6	FSM model of operating modes for sensor node	61
4.7	Principle of node energy simulation in CAPNET-PE tool	63
4.8	The components of wireless sensor node	64
4.9	Block diagram of sensor node in CAPNET-PE tool	65
4.10	Event scenario of energy simulation	66

4.11	Energy consumption versus event rate	69
4.12	Consumption percentage of PAM and FPGA in wireless node	70
4.13	User interface of SPNP tool	71
4.14	TSPN model of sensor node functionality	73
4.15	Failure probability of node components	74
4.16	Comparison of node availability with and without our approach	75
4.17	Structure and reliability diagram of cluster	76
4.18	Comparison of cluster with and without our approach	76
4.19	Principle of PAM functionality	78
4.20	PAM implementation in Microchip microcontroller	79
4.21	IGLOO Nano Starter Kit	82
4.22	PAM design in Libero IDE after connecting components	83
4.23	AD5204 digital potentiometer pin configuration	85
4.24	Characteristics of PTH04070W digital regulator	86
4.25	Block interconnection diagram for adjustable regulator circuit	87
4.26	Adjustable regulator test-bench with PAM implementation in microcontroller	88
4.27	Diagram of PAM block implemented in IGLOO Nano for adjustable regulator	89
4.28	Adjustable regulator test-bench with PAM implementation in IGLOO Nano	90
4.29	Block Interconnection of hardware failure detection platform	92
4.30	Hardware failure detection PCB	93
4.31	Principle of self diagnosis implementation	94
5.1	Structure of sensor cluster	99
5.2	Transmission between the sensor nodes and the Sink	101
5.3	Obstacle impact in communication between cluster and Sink	102
5.4	Cluster-head selection steps	103
5.5	Approach principle deals with energy depletion issue of the Sink	105
5.6	Approach principle deals with hardware rupture issue of the Sink	106
5.7	NED description of FIFO module	107
5.8	NED description of a compound module	108
5.9	C++ codes (.h and .cpp) for programming sensor node behavior	109
5.10	Example of a Tkenv user interface in OMNeT++	110
5.11	Simulation results of network reliability	111
5.12	Simulation results of network lifetime	112
6.1	Reduction percentage of data loss by applying PA-OACH approach	119
6.2	Data throughput improvement by applying PA-OACH approach	120
6.3	Network lifetime improvement by applying PA-OACH approach	121
6.4	Remaining energy of candidate nodes of 2-cluster(a.), 3-cluster(b.) networks	122
6.5	Remaining energy of candidate nodes of 4-cluster(a.), 5-cluster(b.) networks	123

Chapter 1

INTRODUCTION

When dating back to 1950's, the first wireless network named "Sound Surveillance System (SOSUS)" was developed by the United States military, in order to detect and track Soviet submarines. This network used submerged acoustic sensors - hydrophones - distributed in the Atlantic and Pacific oceans. This sensing technology is still being used today in more peaceful applications such as undersea wildlife and volcanic activity monitoring. Up to 1980, the origin of the research in Wireless Sensor Networks (WSNs) called "Distributed Sensor Networks (DSN)" program was started by United States Defense Advanced Research Projects Agency (DARPA), in order to formally explore the challenges in implementing distributed WSNs. DSNs were assumed to have many spatially distributed low-cost sensing nodes that collaborated with each other and operated autonomously, with information being sent to appropriate node for using this information. Through partnering universities such as Carnegie Mellon University and the Massachusetts Institute of Technology Lincoln Labs, DSNs was integrated in academia. Although this program was ambiguous at this time, but WSN technology soon found a home in academia and civilian scientific research.

Recent advances in semiconductor, networking and material science technologies allow ubiquitous deployment of large-scale wireless sensor networks (WSNs). Together, these technologies have combined to enable a new generation of WSNs that offers overwhelming advantages compared to wireless networks developed as 5 to 10 years ago. On the other hand, the use of Wireless Sensor Networks has exploded in the last decades, and is continuing to increase at an ever faster rate. This is changing the way we live, as increasingly people base on wireless connectivity in more and more aspects of daily life.

This introduction aims at presenting the reader with a global outline of our study. Section 1.1 provides a general view on Wireless Sensor Network (WSN) including its hardware structure, operation and communication with physical world. Section 1.2 introduces the need of WSN in human life and its trade market to the year of 2020. And the objectives of this work is presented in Section 1.3. Finally, section 1.4 provides the outline of this thesis dissertation.

1.1 Wireless Sensor Network

Wireless Sensor Networks (WSNs) get a lot of attention in recent years in both academia and industry. The reason is that WSNs get numerous advantages comparing to traditional wired networks due to their low-cost, portability, easy deployment, self-organization, and re-configurability. They may be carried out on everywhere from the human body to be deeply embedded in the environment. Wireless Sensor nodes can easily be deployed in large space with dramatically less complexity and cost compared to wired nodes. Additionally, wireless nodes can self-organize to form routing paths, collaborate on data processing, and establish hierarchies. The WSN is also re-configurable by easily adding and removing sensor nodes. The hardware configuration of wireless sensor node consists of five primary types of resources: computing, storage, communication, sensing and actuating, battery [W. Dargie, 2010 [4]] as shown in Figure 1.1.

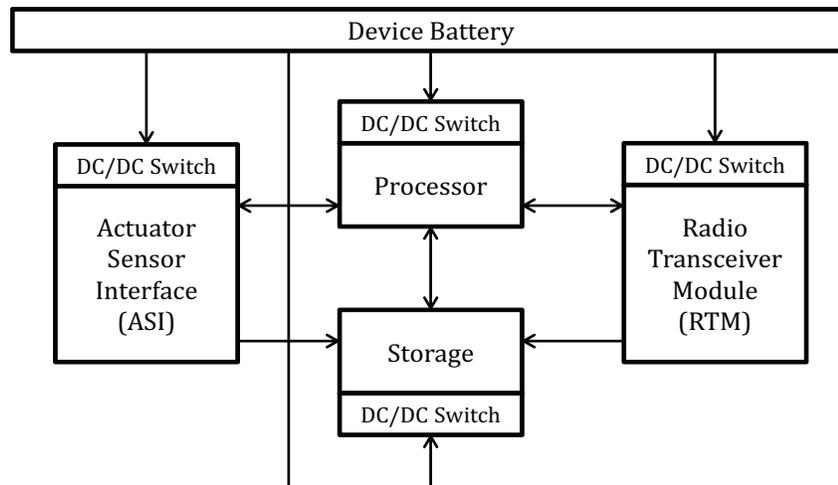


Figure 1.1: Hardware configuration of wireless sensor node

- **Embedded processor**'s functionality is to schedule tasks, process data and control the functionality of other hardware components. The types of embedded processors that can be used in a sensor node include Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) and Application - Specific Integrated Circuit (ASIC). Among all these alternatives, the Microcontroller has been the most used embedded processor for sensor nodes because of its flexibility to connect to other devices and its cheap price.
- **Storage** in a sensor node includes on-chip flash memory RAM of a microcontroller and external flash memory.
- **Actuator Sensor Interface (ASI)** is a hardware device that produces a measurable response signal to a change in physical world, and then provides retro-reaction to control

monitoring environment.

- **Radio Transceiver Module (RTM)** is responsible for the wireless communication of a sensor node. The various choices of wireless transmission media include Radio Frequency (RF), Laser and Infrared. RF based communication fits to most of WSN applications. The operational states of a transceiver are Transmit, Receive, Idle and Sleep.
- **Power Source** is stored in batteries in sensor node. In a sensor node, power is mostly consumed by sensing, communication and data processing, while a little amount of battery energy is dissipated in DC/DC converters. In most WSN applications, more energy is required for data communication than for sensing and data processing.

Wireless sensors link the physical with the digital world by capturing and revealing real-world phenomena and converting these into a form that can be processed, stored, and acted upon. An example of the steps performed by sensor node in data acquisition and actuation is shown in Figure 1.2. Phenomena in the physical world are observed by a sensor device. The capturing electrical signals are often not ready for immediate processing, therefore they pass through a signal conditioning stage. Here, a variety of operations can be applied to the sensor signal to prepare it for signal processing. For example, signals often require amplification (or attenuation) to change the signal magnitude to better match the range of the following analog-to-digital conversion. Then, various filters are applied to the signal to remove unwanted noise within certain frequency ranges (e.g., high-pass filters can be used to remove 50 or 60 Hz noise picked up by surrounding power lines). After this stage, the analog signal is transformed into a digital signal using an analog-to-digital converter (ADC). The signal is now available in a digital form and ready for further processing, storing, or visualization.

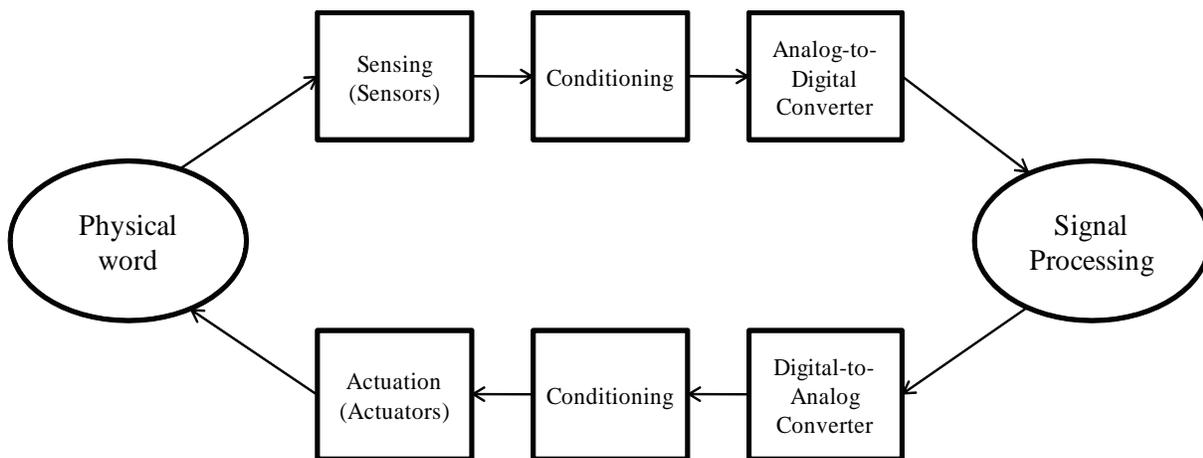


Figure 1.2: Processing steps of data acquisition and actuation

Many wireless sensor networks also include actuators which allow them to directly control the physical world. For example, an actuator can be a valve controlling the flow of hot water, a

motor that opens or closes a door or window, etc. Such a wireless sensor and actuator network (WSAN) takes commands from the microcontroller device and transforms these commands into input signals for the actuator, which then interacts with a physical process, thereby forming a closed control loop as shown in Figure 1.2.

1.2 Trade Market

Large-scale Wireless Sensor Network systems with hundreds wireless nodes are employed in many applications (Figure 1.3) such as:

1. *Infrastructure Security or Counterterrorism Applications*: critical buildings and facilities such as power plants, airports, and military bases have to be protected from potential invasions. Networks of video, acoustic, and other sensors can be deployed around these facilities [Chee-Yee Chong, 2013 [52]]. For example, an initiative in Shanghai Pudong International Airport has involved the installation of a WSN-aided intrusion prevention system on its periphery to detect any unexpected intrusions.
2. *Environmental Monitoring* [D. C. Steere, 2000 [53]] can be used for animal tracking, forest surveillance, flood detection, and weather forecasting. It is a natural candidate for applying WSNs [Chee-Yee Chong, 2013 [52]], because the variables to be monitored, e.g. temperature, are usually distributed over a large region. One example is that Dr. Martinez et al. from the University of Southampton have built a glacial environment monitoring system using WSNs in Norway [K. Martinez, 2005 [40]]. They collect data from sensor nodes installed within the ice and the sub-glacial sediment without the use of wires which could disturb the environment.
3. *Health Monitoring*: WSNs can be embedded into a hospital building to track and monitor patients and all medical resources. Special kinds of sensors which can measure blood pressure, body temperature and electrocardiograph (ECG) can even be knitted into clothes to provide remote nursing for the elderly. Besides, the sensor nodes may be implanted in human body for healthcare purposes [Montón E., 2008 [63]]. This special kind of sensor network called a Body Sensor Network (BSN), allows inexpensive, continuous and ambulatory health monitoring with real-time updates of medical records via the Internet.
4. *Industrial Sensing*: equipment failures cause more and more unplanned downtime when plant infrastructure ages. A statistical report realized by the ARC Advisory Group demonstrates that 5% of production in North America is lost to unplanned downtime. By implementing wireless nodes into machines [Low K. S., 2005, [59]], WSNs make it economically feasible to monitor the "health" of machines and to ensure safe operation. For example, aging pipelines and tanks have become a major problem in the oil and gas industry. Monitoring corrosion using manual processes is extremely costly, time consuming, and unreliable. A network of wireless corrosion sensors can be economically deployed to reliably identify issues before they become catastrophic failures.

5. And a number of others applications in agriculture [Damas M., 2001 [62]], traffic monitoring [Ribeiro A., 2003 [61]], greenhouse control [Serodioa C., 2001 [60]].

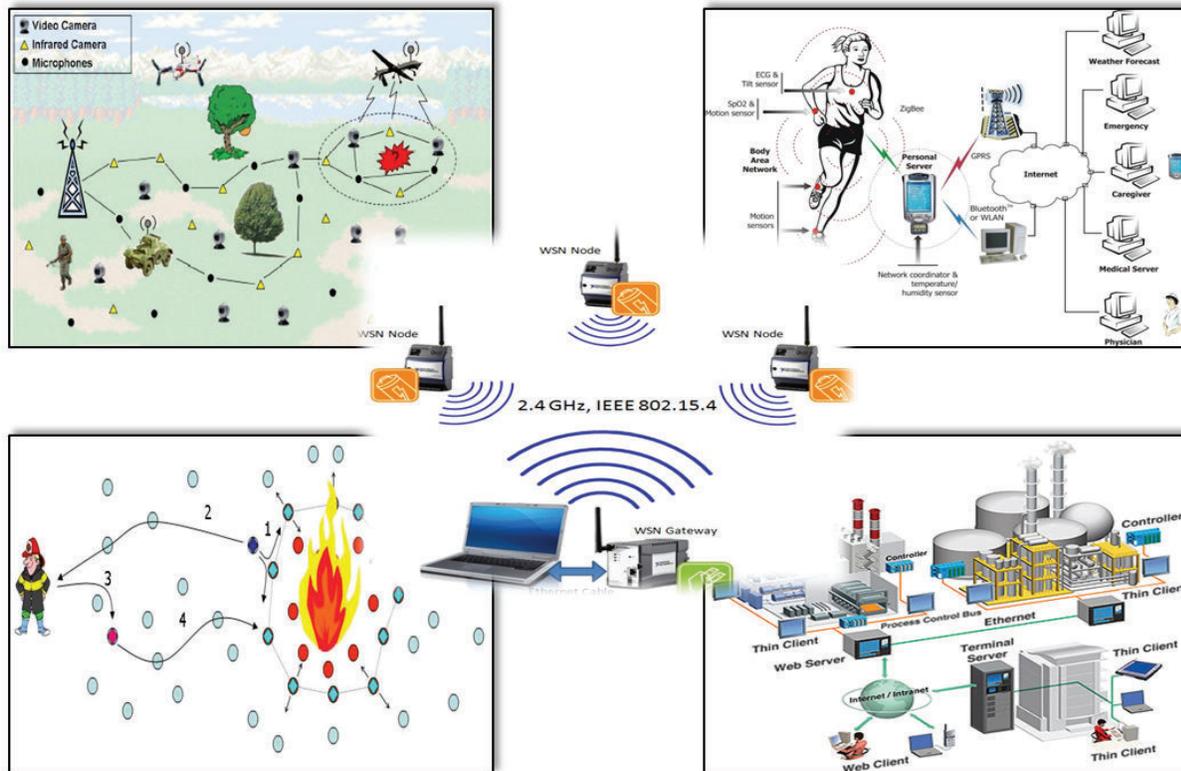


Figure 1.3: WSN applications in daily life

As a result, the global revenue for WSN market continues to exhibit strong growth in recent years. Dr. Peter Harrop [Peter Harrop, 2012 [1]] provides a statistical report of WSN revenue from 2010 to 2014. As seen in Figure 1.4, annual revenue of WSN market increases with stable rate about 30%, particularly the growth rate at 2012 is up to 40%. Besides, WSN trade market is estimated to have a rapid growth from \$1 billion in 2014 to \$2 billion in 2020 with an increasing rate of 200%. That shows massive potential benefits of WSN development in both academia and industry in the future.

1.3 Study Objective

Major breakthroughs in the deep sub-micron technology have led to the emergence of ubiquitous computing [M. Weiser, 1993 [2]]. This technology advance allows to produce small and inexpensive devices like wireless sensor nodes, which may be deployed everywhere to monitor the environment around them. At the beginning of this work, two main questions are raised for me:

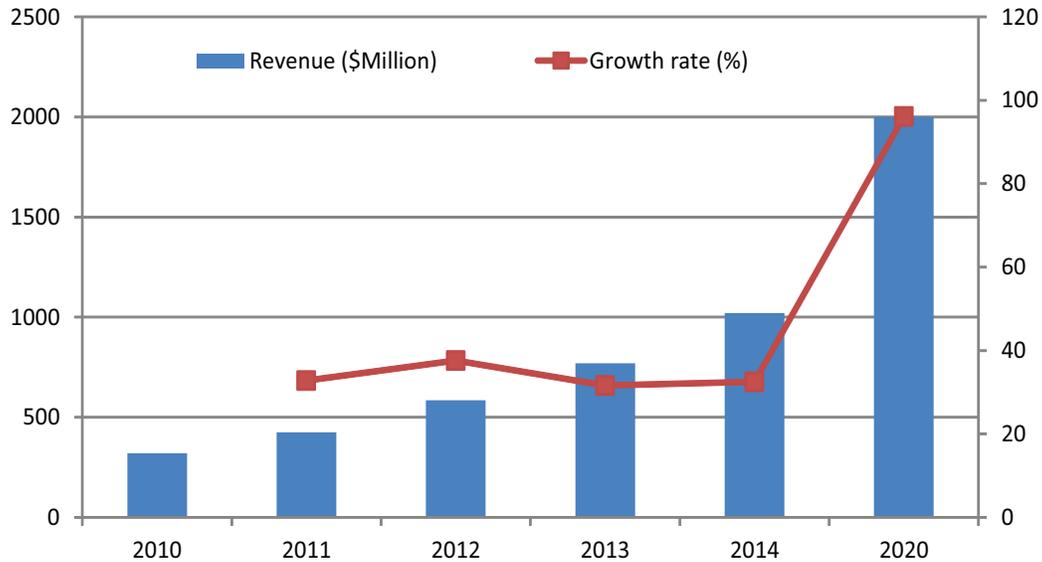


Figure 1.4: Revenue and growth rate of WSN trade market

- "Why energy-efficiency is indispensable for wireless sensor node?"
- "How does wireless sensor node behave in case of instantaneous failure occurred inside its system?"

The uncertain contexts in which recent WSN embedded applications evolve have big impact on these applications. Traditionally, the objective of availability generally doubles hardware and functional redundancy; it means that the overhead is doubled in term of energy and cost. As we know, wireless node system is powered by limited battery; hence power consumption parameter is only set to a number of components and functionalities at minimum resources. However, due to the technology reduction, process variability conducts to increase the possibility of failures. In order to guarantee an acceptable quality of service for the users, and on the operating lifetime of the system, it should carry studies at the upper phases involving both dependability and consumption constraints. Indeed, designing a system by privileging one constraint, then treating the other as a pure parameter would be to consider the second parameter as the restraint of the initial solution and to mask a part of design space. The potential of reconfigurable systems to reorganize during their operation is well demonstrated. These systems are characterized by the concept of architecture (the entire set of resources that can be implemented) and the configuration (how the resources of the architecture are organized temporarily).

The objective of this work is to propose methods and tools to decide how to implement effectively the optimal hardware configuration to address the tradeoff between reliability and consumption. Our work targets WSN applications deployed in the hostile environment where we cannot physically intervene, or difficult due to highly greater maintenance cost overhead

compared to operation cost. Therefore, we aim to provide a configuration employing a variable level of resources (FPGA/processor) to cover the need of dependability and ensure a minimum amount of power consumption. Under autonomous vision, wireless sensor node system is expected to react automatically against failures to keep itself continuing to operate before having technical intervention, and manage effectively energy consumption to extend operating time of sensor node. In the next section, we present a global outline of our contribution to address the above issues.

1.4 Structure of the Thesis Dissertation

The remainder of our thesis dissertation is organized as follows:

Chapter 2: This chapter presents an overview of research context of our study, which describes how the sensor networks are deployed in the environment, and which specifications characterize our targeted networks. Based on this analysis, we list all the critical issues that may occur during network operation. Afterwards, we take a look at the existing works in academic literature to address these issues. These works are inspiration sources for us to do this work. Finally, several novel contributions of our approaches are presented to improve wireless sensor node and network energy-efficiency and availability.

Chapter 3: This chapter presents the basic methods and modeling tool employed in our study. At the aspect of energy management, among the most used methods of energy reduction, Dynamic Power Management (DPM), and Dynamic Voltage and Frequency Scaling (DVFS) are used in our work, in order to prolong the operating time of wireless sensor node. Next, the reconfiguration design is introduced including its principle, failure detection, and decision making. This design method allows increasing the node availability in case of energy depletion or hardware failure. The modeling tool used to model the functionality of wireless sensor node and network is then presented. That allows us to predict all possible faults during network operation, and then take the appropriate solutions to deal with these faults.

Chapter 4: This chapter introduces our novel design approach for wireless sensor node. Our node design is supposed to address all issues related to energy and availability inside sensor node. The methodologies for reducing the energy dissipation and availability improvement for sensor node are then described in detail. Next, our node approach is evaluated with simulation and material implementation, and this chapter ends with several conclusions.

Chapter 5: Since the cluster topology is used for network deployment in our study, this chapter firstly presents used cluster structure. As the WSN structure is various in different applications, the cluster structure specification is crucial to facilitate our work. Then an appropriate cluster-head selection algorithm is proposed to deal with cluster-head unavailability issue, which is described in Chapter 2. We also provide the solution for base station unavailability issue. Finally, our approach is evaluated with simulation to demonstrate its impacts on network reliability and lifetime. This chapter ends with several conclusions.

Chapter 6: This chapter presents global discussions and conclusions based on the rich simulation and implementation results from this work. The actual simulation experimentation and material implementation approach enabled us to identify the crucial factors from hardware and application specifications, which highly determine the efficiency of our solution in energy and availability improvement for wireless sensor node and wireless sensor networks. Finally, our conclusion includes the main contributions of this work and several perspectives are given.

Chapter 2

RESEARCH CONTEXT

The work presented in this thesis is in cooperation with ERYMA Company, and sponsored by Brittany region and General Council (CG56) association. Our industrial partner offers an expertise in material solutions and security system for the prevention, monitoring, maintenance and remote control. The topic on which we are assigned to work is to provide a novel design for wireless sensor networks, in order to improve both energy-efficiency and availability. When WSNs are deployed in hostile environment where the human intervention is very difficult, this design approach provides high autonomy for WSNs by allowing them to find an appropriate solution by themselves, in order to address any issue related to energy depletion or hardware failures. In this chapter, an overview of research context of our study is provided. First of all, we take a look at main topologies that are used to deploy networks in the environment, and then a suitable one is chosen to apply our work. Since there are many different types of WSN applications existing in the trade market, for the sake of simplicity and clarity, the main characteristics of WSN in our study are specified. Based on network specifications, list of critical issues that may occur during network operation are provided. With the help of this list, several solutions will be proposed to address these issues. Next, several reference works that inspire us to do this work are presented here. And finally, some major contributions of our approach are introduced to give a general overview for the readers.

2.1 Network Assumptions

2.1.1 Communication topologies

The supervisors receive data sent from the sensor nodes of the network through a base station called Sink node. The communication link between the Sink and the source nodes is followed by two basic types: single-hop (or centralized) and multi-hop (or decentralized) communication topologies. The Star topology is basically single-hop communication in which the data being sensed by a node in the network is directly sent to the Sink (base station) node, where the supervisors can access data. This protocol is possibly optimal method of communication if the Sink is close to the sensor nodes. From [W. R. Heinzelman, 2000 [44]], the required energy to

transmit a k -bit data packet on a distance d using radio link is computed as follows:

$$E_{Tx}(k, d) = E_{elec} * k + \alpha_{amp} * k * d^2 \quad (2.1)$$

And to receive this data packet, the radio module expends:

$$E_{Rx}(k) = E_{elec} * k \quad (2.2)$$

Where E_{elec} is the required energy per bit to run the radio circuitry and α_{amp} is amplifier factor per square meters to transmit a bit. From Eq.2.1, the further the distance from sensor node to the Sink, the more consuming their communication. Since wireless nodes are powered by limited battery, this topology reduces the space coverage of the network. When the number of nodes inside the network increases, the buffer overflows will appear at Sink node if many sensor nodes try to communicate simultaneously with the Sink. To address these issues, the cluster topology based on multi-hop communication is proposed to subdivide the network into sub-networks called clusters, and each cluster has its own cluster-head called Gateway node that gathers the data from the nodes in its cluster, and then sends it to the Sink as depicted in the Fig.2.1.

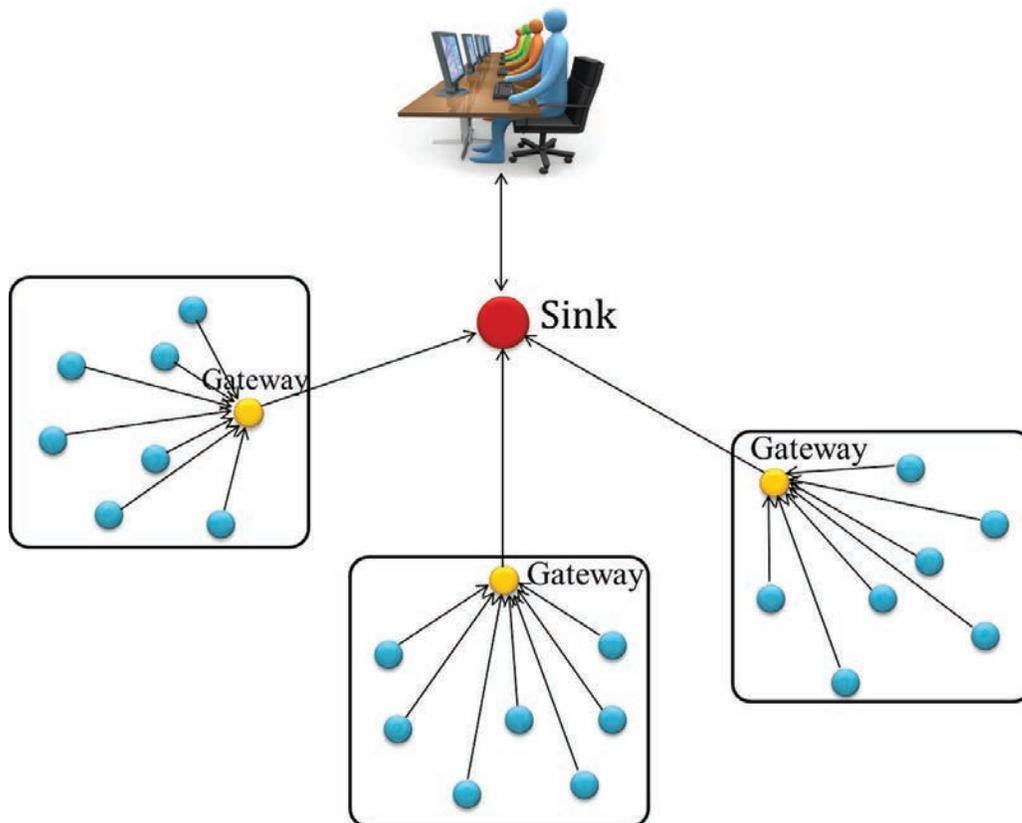


Figure 2.1: Structure of cluster topology

As seen in this figure, the sensor nodes are now communicating data over smaller distance in the cluster environment, which reduces their transmitting energy. Therefore, using the cluster or decentralized topology allows extending the space coverage of monitoring environment and to mitigate the problem of buffer overflow at Sink node. Briefly, the centralized methods are accurate and efficient for small networks, but they are not appropriate for large-scale networks. Since large-scale networks are targeted in our study, the cluster topology is used in the remaining chapters.

2.1.2 WSN characteristics

The network used in our study has the following characteristics:

- The network consists of n clusters that are fixed. Each cluster contains a constant number of nodes including one Gateway and m source nodes.
- The Sink node is fixed at a far distance from the clusters.
- The source nodes communicate directly with the Gateway of their cluster, and all the Gateway nodes have direct communication with the Sink.
- All the nodes are immobile (i.e. their position is fixed), homogenous, and energy constrained with uniform energy, except the battery of Gateway nodes has a bigger storage capacity of energy because these are critical elements.
- All the sensor nodes are able to harvest energy (sun, wind) from environment.

The *Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)* protocol [Wei-Ye, 2002 [51]] is used for radio communication in the network. As shown in Figure 2.2, when the radio channel is available, sensor node sends the *Request To Send (RTS)* message to destination node, if it receives the *Clear To Send (CTS)* message, it will transmit the data packet to destination node. Otherwise, sender node resends RTS message after a back-off time. This allows avoiding the collisions of data transmission in the network. After forwarding the data packet to destination node, if sender node does not receive any *acknowledgement (ACK)* message during an interval of time, it will resend the same data packet.

After specifying the main characteristics of target networks, a list of main issues that may have negative impacts in network lifetime and availability is provided in next section.

2.2 Problem Statement

Since the design of WSN architecture contains a vast challenge in various domains, hence, the objective of this work is to introduce a generic approach for WSN that mainly involves two aspects: energy-efficiency and availability, in order to extend network lifetime and reliability.

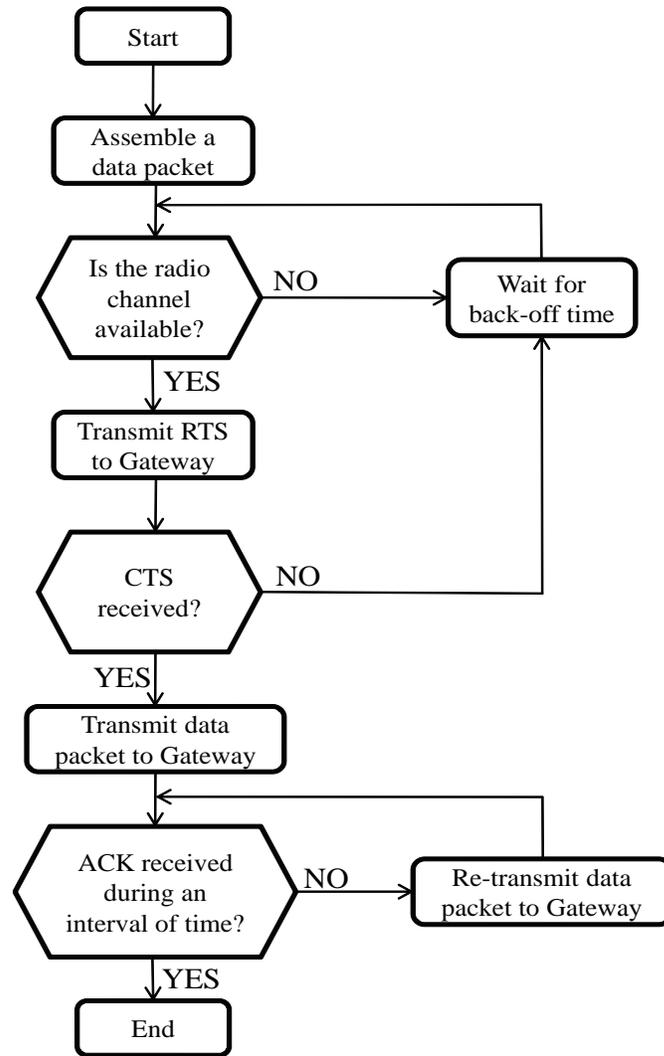


Figure 2.2: Diagram of CSMA-CA communication protocol

Based on the decentralized structure of WSN, we propose to decompose the issues in three levels of hierarchy, as summarized in Tab.2.1.

At the node level, since wireless nodes are battery-powered devices, an inefficient use of energy conduces to quick depletion of node battery, and consequently, node lifetime is reduced. After a long operating time, several nodes are dead due to energy depletion, which dramatically degrades the accuracy and reliability of WSNs. Besides, nowadays wireless sensor nodes are usually deployed in harsh environments such as Tsunami Detection [K. Casey, 2008 [36]], Hazardous Gas or Intrusive Detection [N.Ferry, 2011 [26]], Fire Detection in Forest, etc., to realized short-term or long-term missions. This kind of environment, which contains a lot of implicit hardware failure issues, makes faults occur frequently in WSNs, particularly in wireless nodes. The faults may range from failing components of sensor node [F. Koushanfar, 2002 [38]] includ-

Table 2.1: Some key issues of WSN in our study

Hierarchy level	Issue	Consequence
Network	* Sink stops to operate due to hardware failure	* Communication disruption with whole network
Cluster	* In case of out of control Gateway, a bad node is promoted to cluster head	* Decrease of packet delivery ratio * Wasted energy for re-transmitting data * Throughput degradation
Node	* Ineffective use of battery energy	* Quick depletion of battery energy
	* Hardware failure of node components	* Unavailability of sensor node

ing five primary types of resources: computing, storage, communication, sensing and actuating, to completely crash fault of sensor node [M. Yu, 2007 [37]]. Since faults inside sensor node are unavoidable and unpredictable, consequences can be severe in term of human life, environment impact or economic loss, hence it is important to determine promptly if a node component is whether still in operation or failed. That allows sensor node react consistently against occurred hardware failure issues to make itself less vulnerable. Furthermore, when WSNs are deployed in harsh environments, immediate human intervention is very difficult. Under autonomous vision, wireless nodes are expected to detect and identify the failed components, and then execute node reconfiguration to find corrective solution before having technical intervention. That stresses the need of hardware failure self-diagnosis to improve the availability of wireless sensor node, which keeps the sensor network working as expected.

At the cluster level, a major issue is raised to be the Gateway unavailability, as we will loose the data from all the nodes of a cluster if their Gateway ceases to operate due to energy depletion or hardware failure. Thus, an appropriate algorithm is needed to find alternative cluster-head, which can replace the Gateway during the recharge of its depleted battery or its reparation. This mechanism is very important because if a bad node is promoted to cluster-head role, this will diminish packet delivery ratio due to communication rupture between cluster and the Sink. As a result, the dissipated energy for data re-transmission is increased, and the throughput is degraded.

At network level, the Sink is the most critical device, if it is down, the supervisor will loose the whole connection with the network that stresses a consistent solution to fix this issue.

2.3 Reference Works

2.3.1 Energy-efficiency

At node hierarchy, many existing energy reduction mechanisms for WSNs target different layers:

Communication layer (MAC protocol) – Time Division Multiple Access (TDMA) protocol [K.Arisha, 2002 [13]] [V. Rajendran, 2003 [14]] is used for network communication where each sensor node only needs to listen and broadcast for its own time slots, and its radio transceiver

is turned off in the rest of time to reduce energy transmission. However, the disadvantage of this protocol is that the dead time between time slots limits the potential bandwidth of a TDMA channel. *Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)* protocol described in [Tijs-Van Dam, 2013 [15]] [Mo-Sha, 2009 [16]] avoids collisions of data transmission in the network, in order to reduce energy consumption due to data retransmission. In this protocol, sensor node sends data to the base station when the radio channel is idle. Contrary to TDMA protocol, the CSMA-CA allows sensor nodes to transmit their data whenever something has to be sent. Both TDMA and CSMA-CA protocols allow reducing energy transmission due to data collision, idle listening, and overhearing.

Network layer – Two major strategies of energy-saving in this layer is *Power-Aware Routing* and *Maximum Lifetime Routing*. The first strategy [Hossam-Hassanein, 2006 [17]] [Lopez-Gomez, 2009 [18]] aims to find a route that consumes the least possible power, and the second strategy tries to balance energy dissipation among sensor nodes to prolong the operation lifetime of sensor network [C. Srisathapornphat, 2002 [19]].

Transport layer – *Transmission Control Protocol (TCP)* described in [V. Tsaoussidis, 2000 [20]] is designed to reduce the retransmission in the network. ATCP and TCP-Probing [J. Liu, 2001 [21]] can adjust TCP's retransmission strategy to reduce unnecessary retransmissions, in order to minimize power consumption and achieve higher throughput. An experimental study in [S. Agrawal, 2002 [22]] improves TCP performance, which allows sensor network to efficiently use energy.

Physical layer – The turn-off technique like *Dynamic Power Management (DPM)* [M.T. Schmitz, 2004 [3]] is employed in WSNs to reduce energy dissipation [A. Sinha, 2001 [23]]. The strategy of this method is to turn off a part of the circuit or run it in degraded mode (sleep or deep sleep mode), which reduces consumption. However, it requires suitable wake-up times that do not violate the time constraint for the application. The second technique called *Dynamic Voltage Frequency Scaling (DVFS)* [M.T. Schmitz, 2004 [3]] is used to reduce the energy consumption of processing elements by adjusting their supply voltage and operating frequency [M.K.Bhatti, 2010 [31]]. Nevertheless, the downside of this technique is the overheads in term of energy and switching time between supply voltage levels. Finally, the effective use of both two techniques in WSNs is still missing.

2.3.2 Availability

Availability on the service level means that the service delivered by a WSN (or part of it) is not affected by failures and faults in underlying components such as single node or node subsystem [L. M. S. D. Souza, 2007 [39]]. As the availability issue in wireless sensor network is classified in three hierarchies in our study, a searching in academic literature focused on how to deal with this issue in each hierarchy layer.

Node layer

A sensor node can get subject to hard or soft faults, and based on duration, faults can be considered as permanent, intermittent, or transient. Mahapatro A. and Pabitra M.K. [Mahapatro A., 2013 [41]] provide a full description of all categories of faults being assumed to exhibit in sensor node with respect to fault diagnosis. They classified faults in seven categories such as crash, omission, timing, incorrect computation, fail-stop, authenticated Byzantine, and Byzantine. When crash fault is considered as permanent and hard fault, the remaining faults are soft faults and may be permanent, intermittent or transient. These faults make the challenge in scientific research, therefore many fault detection and diagnosis techniques have been proposed for WSNs. To facilitate the readers to understand each technique, Arunanshu and Pabitra present a technique-based taxonomy framework (as shown in Fig.2.3) that categorizes these techniques developed for WSNs.

As discussed in [Mahapatro A., 2013 [41]], in centralized approaches, the network availability is managed by a central node that is equipped with high computational power, larger memory size and uninterrupted energy sources. The centralized methods are accurate and efficient for small networks, but they are not appropriate for large-scale networks. Two main reasons are following:

- The Sink or base station must carry out a bulky amount of data sent from all sensor nodes in network for analysis and diagnosis to detect any failures occurred, which is very expensive in term of time and resources (computing and storage).
- The header nodes called cluster-heads, which are responsible for the communication between clusters and the Sink, have problem of quick energy depletion due to extremely high data transmission for centralized fault diagnosis. That leads to a critical issue because the Sink will loose connection with remaining nodes in cluster if its cluster-head runs out of energy.

For these reasons, the distributed approaches are proposed to overcome the limitations of centralized approaches. The distributed approaches allow every sensor node to monitor its operating state, and then to make decisions at certain level before sending information to the Sink. The more decisions a sensor node can make, the less information is sent to the Sink. Therefore, we focus on distributed diagnosis approach in our work. Our objective aims to present a novel diagnosis technique belonging to node-self detection category as shown in Fig.2.3. In this category, the node system is capable of detecting its own operating state by using additional hardware to sensor node architecture. When making an overview of academic works, many approaches belonging to node-self detection category have been carried out to deal with hardware failures in wireless sensor node.

S. Harte et al. [S. Harte, 2005 [32]] propose an approach where a node would perform a self-diagnosis based on the measurement of a number of miniature accelerometers mounted on

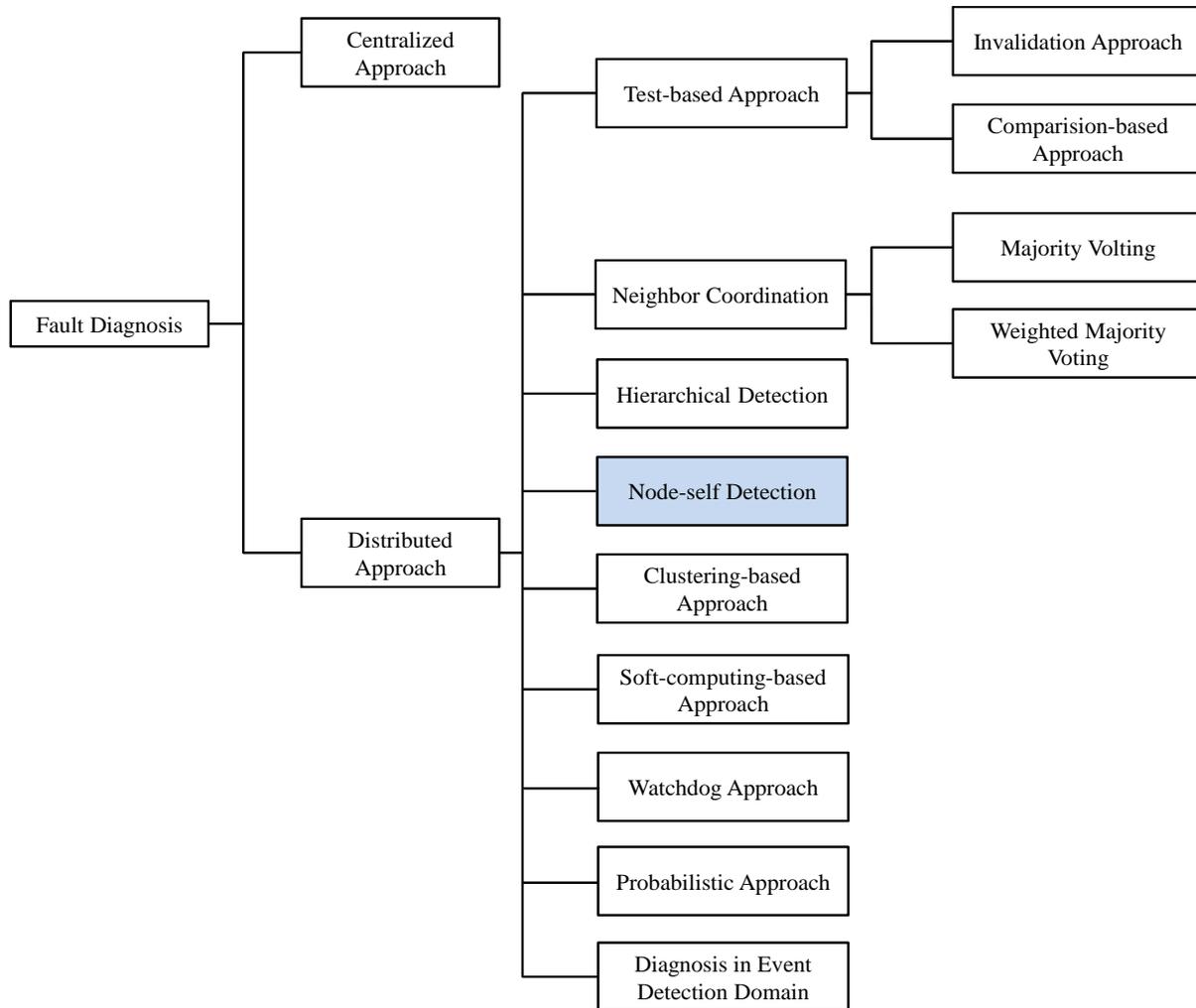


Figure 2.3: Taxonomy framework for fault detection techniques

board, in order to determine if a node suffers from an impact that could lead to hardware malfunctions. In this technique, damaged accelerometers are covered by some redundancy added in sensor node. However these accelerometers are always turned on and their measurement data are continuously analyzed, which increases a lot of energy dissipation. Besides, this solution occupies a substantial area in sensor node board.

Mahapatro and Khilar [A. Mahapatro, 2012 [33]] [A. Mahapatro, 2013 [34]] present a self-diagnosis approach to identify hard and soft faults in CMOS image sensor node by using a distributed algorithm. In this technique, the sensor node is considered as soft fault-free if the sensor reading agrees with readings of more than a number T_k of 1-hop neighbors. A time mechanism is used to detect hard faults where an unreported node is detected as hard faulty. This CMOS image sensor architecture uses Reed-Solomon codes to identify and correct errors in transmission. Since the data of sensor reading are compared to those of a number $T_k > 1$

neighboring nodes, as a result the energy overhead is high due to the radio communication.

Benini et al. [L. Benini, 2000 [35]] introduce an abstract discrete-time model for complete power supply sub-system that could estimate the lifetime of the battery by analyzing the battery discharge curve, in order to avoid the faults caused by battery exhaustion.

Koushanfar et al. [F. Koushanfar, 2002 [38]] propose self-diagnosis of sensor nodes in WSNs. This approach observes the binary outputs of its sensors by comparing with the predefined fault model. Since data diagnosis is continuously run, the energy overhead is expensive. Additionally, this method occupies processor during data diagnosis.

We aim to provide a novel low-power, low-cost, easy-implementation, and accurate hardware failure detection and localization in this work. Our approach also enables material re-configuration in case of failure that is still missing in previous works, in order to offer self-reparation ability for sensor node. As a result, the node availability is significantly increased and the maintenance cost is highly reduced.

The next sub-section provides a general overview of existing works for the availability issue at cluster hierarchy layer.

Cluster layer

Since the Gateway unavailability is a critical issue during network operation when employing cluster topology, many approaches are proposed to deal with this issue. Salhie et al. [A. Salhie, 2001 [50]] presents an extended version of cluster topology called Mesh topology. In this topology, the nodes of one cluster are able to communicate with the nodes of other clusters (as depicted in Fig.2.4). Therefore, in case of Gateway death due to energy depletion or hardware failure, the nodes may still send their data to the Sink via other clusters. However, Mesh topology contains several major drawbacks such as high latency of data propagation, complexity increase in data routing table control, and high energy overhead. For these reasons, many researchers focused on cluster-head selection approaches inside the cluster to improve cluster availability.

The LEACH protocol presented in [W. R. Heinzelman, 2000 [44]] allows randomized rotation of cluster-head role in the cluster. At the beginning of each round, the decision of cluster heads is made by each node n choosing a random number in the interval $[0, 1]$, if this number is less than the threshold $T_n(t)$ which is computed using Eq.2.3, the node becomes a cluster head for the current round.

$$T_n(t) = \begin{cases} \frac{P}{1 - P \bmod \frac{1}{P}} & \text{if } n \in G \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

Where P is the desired percentage of cluster heads, r is the current number of rounds, and

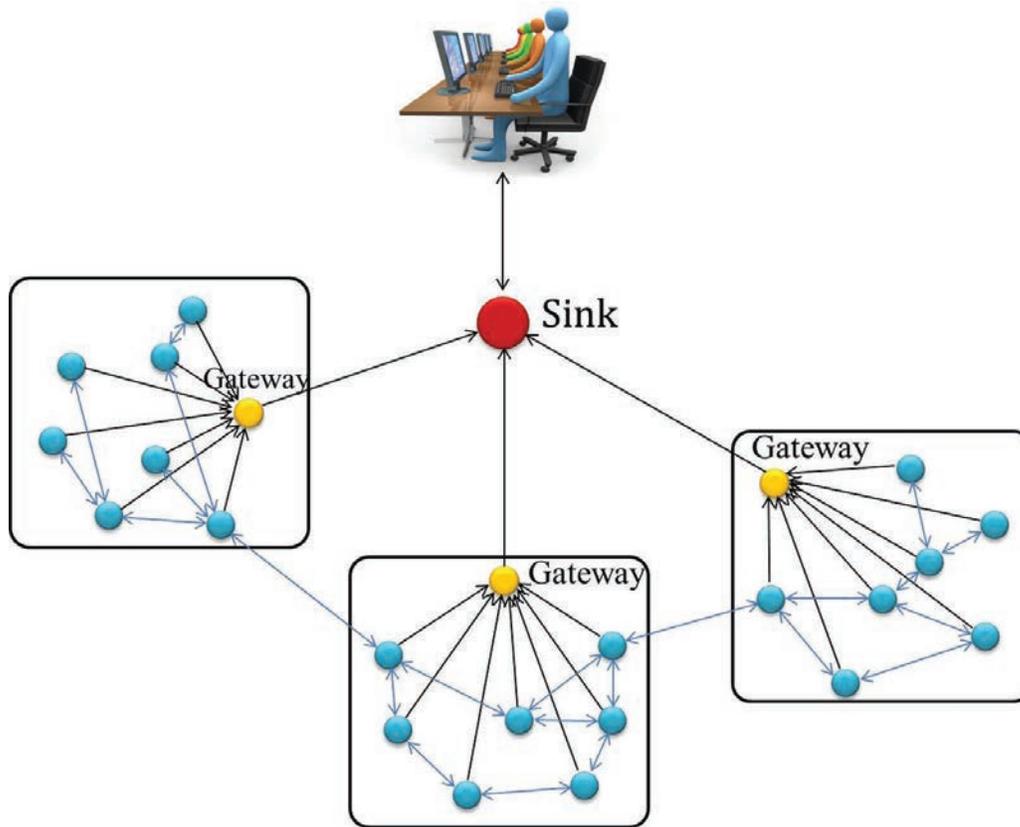


Figure 2.4: Mesh topology for network deployment

G is the set of nodes that have not been cluster-heads in the last rounds. The nodes that are cluster-heads in the previous rounds cannot be cluster heads for the next rounds. After $\frac{1}{P} - 1$ rounds, $T = 1$ for any nodes that have not yet been cluster-heads, and after $\frac{1}{P}$ rounds, all nodes are once again eligible to become cluster-heads. However, the Eq.2.3 shows that all non cluster-head nodes can elect themselves to be Gateway no matter how little residual energy they have. Therefore, if a node with small remaining energy is selected, its battery will quickly be drained implying reliability and lifetime problems for the network. As a result, it does not guarantee an even distribution of energy among all the nodes in the network.

LEACH-DCHS in [Y.H. Liu, 2008 [45]] and A-LEACH in [A.M. Solaiman, 2008 [46]] suggest the improvement of LEACH by including remaining energy into cluster-head selection algorithm. These protocols make some modifications to the threshold computation, which allows the node with more residual energy has a bigger probability to become cluster-head. However, another issue is raised when the nodes locating very far from the Sink are selected for cluster-head role, their battery will draw quickly due to large energy dissipation of data transmission.

EACHS-LEACH in [Y. Liang, 2005 [47]] and M-LEACH in [Y. Liu, 2009 [48]] protocols include additional factor of distance to Sink, and their simulation results show that this strategy of cluster heads election achieves great advance. There are a lot of other works as listed in [K. Ramesh, 2011 [49]] that make a significant improvement compared to basic LEACH protocol.

In our study, the wireless networks are deployed in harsh environment in the applications such as fire detection in forest, hazardous gas detection, etc.; if some temporary obstacles appear instantaneously between the future cluster head and the Sink, this will interrupt their radio communication. As a result, the packet delivery ratio and throughput of the network are diminished. Additionally, the dissipated energy for transmission is massively increased due to data re-transmission. However, this aspect is still missing in existing works.

Network layer

The network layer relates to the availability of base station or Sink node. In our study, the path between the Sink node and supervisor is strictly monitored by the supervisors because it is critical element. Therefore, we assume that there is no obstacle in this path. In all existing works, Sink node is powered by no-limited energy budget and is considered to have strong immunity against failure, hence there is no paper considering this issue.

2.3.3 Conclusion

After getting a look at existing works in academic literature, a summary is made to illustrate their location in technique classification as depicted in Fig.2.5. As seen in this figure, there are few works considering both aspects of energy and availability, even no work considers these two aspects for both wireless sensor node and network. That motivates us to propose a novel design approach with limited material redundancies for managing effectively energy consumption, and improving availability for both wireless sensor node and network. These two aspects are attached to each other in our approach:

- Increasing energy-efficiency leads to improve node availability related to energy depletion.
- Based on consumption monitoring, our self-diagnosis method can detect and localize accurately any hardware failure inside sensor node. Then it chooses an appropriate solution by running reconfiguration.

2.4 Novel Contributions

This work presents a novel architecture for managing energy consumption and availability for both wireless sensor node and network layers with limited redundancies, and its main contributions are:

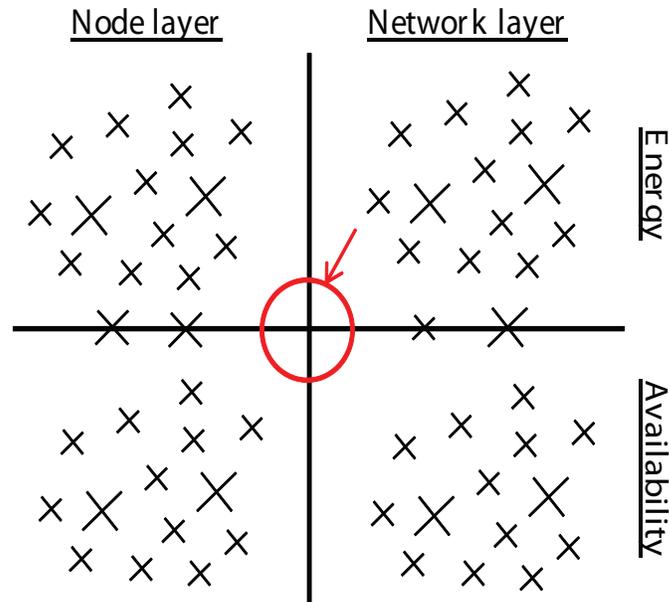


Figure 2.5: Classification of energy and availability approaches for WSN

- A Power and Availability Manager (PAM) is integrated into the system of sensor node layer. This component is responsible for selecting the suitable operating mode for sensor node based on arrived events at discrete time. On one hand, energy budget of sensor node is used effectively that leads to extends battery lifetime. On the other hand, component availability is supervised by this component.
- A low-cost, low-power and accurate self-diagnosis method is proposed to deal with any hardware failure issue in the node layer. If a component is failed, the node system will automatically detect and localize it, and reconfiguration decision will be made promptly to keep sensor node still operating.
- At network design, since cluster topology is employed in our study to deploy the network, a hierarchical structure is proposed for every cluster. Getting inspired from LEACH protocol, this work also proposes new ways of electing a new cluster-head to replace the Gateway node, when this node is out of control due to energy depletion or hardware failure. Next two main criteria remaining energy and distance to the Sink node, the elected cluster-head is also based on obstacle-aware criterion. Accordingly, the network throughput and lifetime are significantly improved.

Chapter 3

BASIS BACKGROUND

This chapter introduces the basic background for reducing energy consumption and improving availability. As explained previously, our work bases on two major constraints of energy and dependability for designing wireless sensor node. The plan of this chapter is as follows: these major constraints and their impacts are firstly introduced. Next, the methods we based on to manage effectively energy consumption is presented. Then the detailed description of reconfiguration steps and the modeling tools for discrete event system (DES) are presented. Finally, some conclusions are given.

3.1 Constraints and Impacts

3.1.1 Ineffective use of energy

Major challenge in wireless sensor networks is the lack of energy efficiency in the sensor node and network, and as a result shortening network lifetime due to the failure of the sensors as they have limited energy. Therefore, on one hand sensor node is required to autonomously manage energy consumption. On the other hand the network must distribute evenly energy load among the nodes in each cluster. These conditions allow effective use of battery energy that is very crucial to extend the operating time or availability of wireless sensor node, which strengthens the network reliability.

3.1.2 Hardware failures

When being deployed in hostile environments that contain lots of implicit hardware failures, human intervention is very difficult in case of hardware incident due to highly greater maintenance cost overhead compared to operation cost. This constraint has a big impact on network dependability because it probably decreases the node availability. Consequently, one or several sensor nodes cease operating that conducts to reduce the coverage space, or the reliability and accuracy of the network. The most usual solution is to double the node redundancy, i.e. the number of sensor nodes in the network. However, this solution is very expensive because it

doubles the required energy battery and WSN cost. That stresses the need of reconfiguration ability of sensor node that allows it to react promptly against any hardware-failure. The cost and energy overheads of this solution are obviously much lower than the usual solution above, which motivates us to add reconfiguration ability by adding some low-cost and low-power material redundancies in novel design of wireless sensor node.

3.2 Methods for Power and Energy Management

Complementary Metal-Oxide-Semiconductor (CMOS) circuit is used the most to fabricate complex integrated circuits such as microprocessors, microcontrollers, static RAM, and other digital circuits. One of the main important characteristics of this circuit is low static power consumption compared to previous logic circuits. However, the rapid growth in semiconductor technology has led to shrinking of feature sizes of transistors using deep submicron (DSM) process. That keeps on increasing the transistor density in single chip to fabricate very large scale circuits with complex functionality. As a result, the power dissipation does not cease increasing, and becomes a vital design metric for current battery operating devices like wireless sensor node. The next sub-section presents the main sources of energy dissipation in wireless node circuit.

3.2.1 Sources of energy dissipation

In wireless node circuit, the energy is consumed mostly by the components such as microprocessor, static RAM, sensors, actuators and transceiver radio module. There are two kinds of power dissipation in these CMOS components: dynamic and static [M.T. Schmitz et al., 2004 [3]]. The static or leakage power in CMOS circuit arises from the leakage currents flowing through the transistor when there are no input transitions and the transistor has reached steady state. The dynamic power in CMOS circuit dissipates only when there are logic switching activities within the circuitry. Thus, the total power dissipation of whole node circuit is given by:

$$P_{tot} = P_{static} + P_{dynamic} \quad (3.1)$$

The static or leakage power in CMOS circuit arises from the leakage currents flowing through the transistor when there are no input transitions and the transistor has reached steady state. The leakage power includes three main sources: sub-threshold leakage current, tunneling current through gate oxide and leakage current through reverse biased diodes. The sub-threshold leakage current is the drain-to-source leakage current when the transistor is OFF. This happens when the applied voltage V_{GS} is less than the threshold voltage V_{TH} of the transistor. This leakage current is computed by the following formula:

$$I_{SubThreshold} = \mu \cdot C_{OX} \cdot V_{th}^2 \cdot \frac{W}{L} \cdot e^{\frac{V_{GS} - V_{th}}{n \cdot V_{th}}} \quad (3.2)$$

Where μ denotes carrier mobility, C_{OX} is gate capacitance, $V_{th} = \frac{K \cdot T}{q}$ where K is Boltzmann constant, T is absolute temperature, q is electron charge, W and L are the width and length

of the transistor. Several techniques are presented in the survey [B.S. Deepaksubramanian et al., 2007 [30]] that are applicable to reduce sub-threshold leakage current for portable devices.

Contrary to sub-threshold leakage current, the tunneling current and reverse biased leakage current occur when the transistor is ON. The tunneling current dissipates when very small thickness level electrons tunnel across the very thin insulation. Tunneling current becomes very important for transistors below 130 nm technology. Meanwhile the reverse biased leakage currents are formed due to formation of reverse bias between diffusion regions and wells, wells and substrate. In modern CMOS circuits, this dissipation is very small compared to the sub-threshold leakage and tunneling currents.

The dynamic power in logic design increases whenever logic toggles from 1 to 0 or from 0 to 1. The dynamic power occurs because of two main components: short-circuit power P_{sc} and switching power P_{sw} . The short-circuit power dissipates during finite rise/fall time of transistor transition. For example, from off to on, both the transistors will be on for a small time interval in which current will find a path directly from V_{DD} to ground. Meanwhile, the CMOS circuits dissipate the switching power by charging and discharging of the effective circuit load capacitance. A simple gate-level circuit is shown in Fig.3.1. This inverter undergoes the following transitions. First, the input signal y is set to high(1), the transistor $Tr1$ is open (not conducting) when the transistor $Tr2$ is conducting. Accordingly, the capacitance C_{load} is discharging since $Tr2$ pulls the capacitance to ground. Now consider a transition from high(1) to low(0) at the input signal y , the transistor $Tr1$ is conducting when the transistor $Tr2$ is open. Thus $Tr1$ connects the capacitance C_{load} to the supply voltage V_{dd} source, charging C_{load} via i_C . As cited in [M.T. Schmitz et al., 2004 [3]], the switching power is the dominant one that accounts approximately more than 95% dynamic power consumption. In our work, we concentrate on the methods to reduce the static and dynamic or switching powers when the transistors are on.

Before introducing the methods we used for reducing the energy consumption, some notions are given. The Fig.3.2 depicts an example of the task graph, task mapping and task scheduling in the architecture of wireless sensor node. The task graph consists of three tasks: T_1, T_2, T_3 (as seen in Fig.3.2a). These tasks are implemented in node processor as illustrated in Fig.3.2b.

From the task graph (Fig.3.2b), the task scheduling is depicted in Fig.3.2c, in which the *idle times* and *slack times* are defined as follows:

- *Idle time* refer to periods in the scheduling when MCUs do not experience any workload, during these intervals the elements are redundant.
- *Slack time* is the difference between task deadline and the task finishing time of sink tasks (tasks with no outgoing edges, i.e., slack times are a result of over-performance).

The next sub-sections introduce two main methods that will be applied in our approach to reduce the energy consumption of the sensor node.

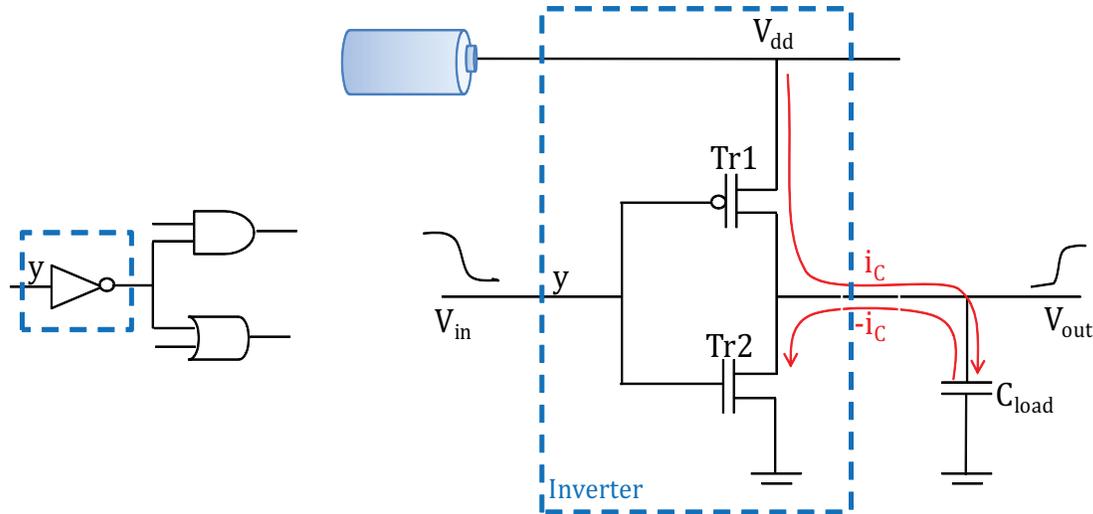


Figure 3.1: Switching power dissipation in an inverter circuit

3.2.2 Dynamic power management

The strategy of Dynamic Power Management (DPM) is to turn off a part of the circuit or run it in degraded mode (sleep or deep sleep modes) when no workload is experienced. An advantage of DPM policy is to be able to be applied not only for digital circuit, but also for other system elements such as displays, hard drives, and analog circuits. This policy is well-known from the advanced power management (APM) and widely used in all electronic devices in order to reduce the energy dissipation.

However, the restart of an element requires a time interval and power overhead to restore its fully functional state. Therefore, suitable wake-up time and power overhead are necessary for this method in order to do not violate the time constraint and achieve the highest possible power savings.

3.2.3 Dynamic voltage and frequency scaling

To understand the principle of the Dynamic Voltage and Frequency Scaling (DVFS) technique, we review the example of logic inverter circuit as depicted in Fig.3.1. As explained previously, the dynamic or switching power of this inverter circuit dissipates due to the logic input transition from high(1) to low(0), where the transistor $Tr1$ is conducting when the transistor $Tr2$ is open. Thus $Tr1$ connects the capacitance C_{load} to the supply voltage V_{dd} source, charging C_{load} via i_C . Obviously, the transition from low(0) to high(1) does not draw any current from the supply voltage source. This indicates that power from the battery is only dissipated during input transition from high(1) to low(0), when the load capacitance is charged. The power dissipation by transition from 1 to 0 is given by:

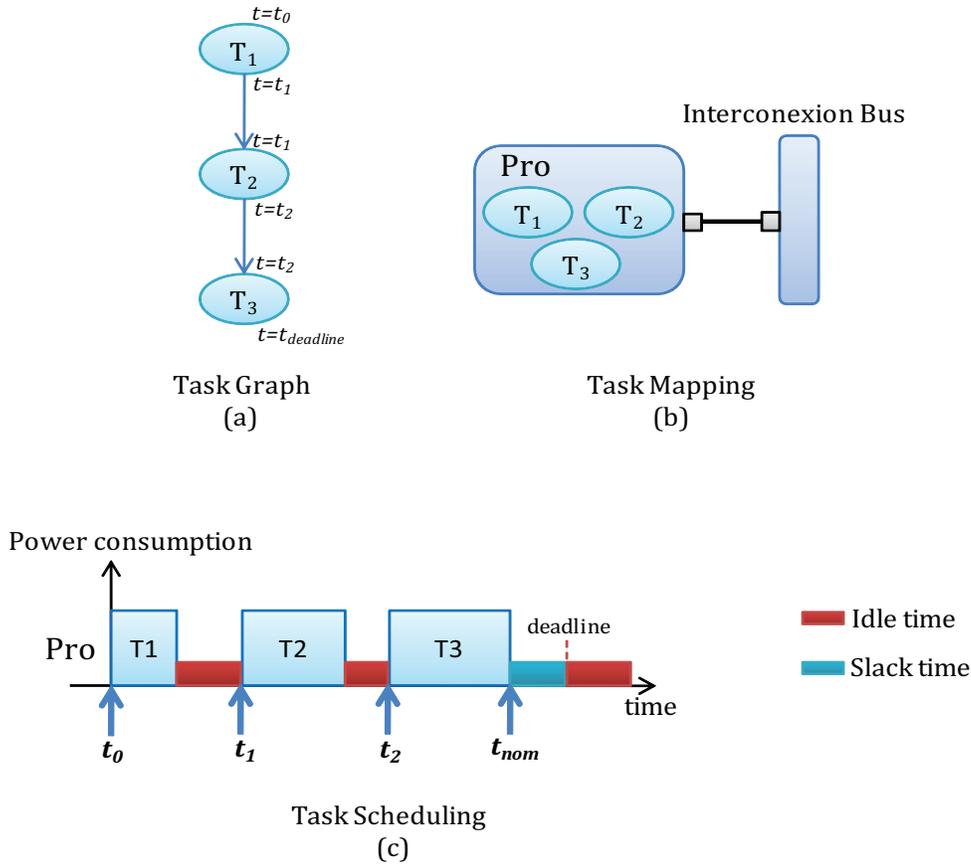


Figure 3.2: Task graph mapping and scheduling example

$$P_{sw} = V_{dd} \cdot i_C \quad (3.3)$$

Where the charging current i_C changes according to the dynamic voltage on the output:

$$i_C = C_{load} \cdot \frac{dV_{out}}{dt} \quad (3.4)$$

The dissipated switching energy of one clock cycle, which takes a period time of T , can be calculated as:

$$E_{sw}^{0 \rightarrow 1} = \int_0^T P_{sw} dt = V_{dd} \cdot \int_0^T i_C dt = C_{load} \cdot V_{dd}^2 \quad (3.5)$$

Where the period time is depended on the operating frequency f at which the circuit is

clocked. The total energy E_{sw}^{ta} drawn from the battery for performing the task t_a depends additionally on the number of clock cycles N_C needed to execute this task, and the switching activity α . Therefore, this total energy dissipation is computed:

$$E_{sw}^{ta} = N_C^{ta} \cdot \alpha \cdot C_{load} \cdot V_{dd}^2 \quad (3.6)$$

Dividing Equation 3.6 by the execution time of the task $T \cdot N_C^{ta}$, we have the equation for power dissipation due to switching as:

$$P_{sw}^{ta} = \alpha \cdot C_{load} \cdot V_{dd}^2 \cdot f \quad (3.7)$$

As seen in the Equ.(3.7), decreasing the operating frequency leads to a reduction in power dissipation, but it does not reduce the energy consumption (Equ.(3.6)), which is important for battery-life. For example, considering a computation that requires 10ms on PE (processing element) running at 10MHz and dissipating 200mW. This computation requires an energy of $200\text{mW} \cdot 10\text{ms} = 2\text{mJ}$. If the operating frequency is reduced from 10MHz to 5MHz, the power dissipation decreases to 100mW, according to the Equ.(3.7). Nevertheless, the frequency reduction increases the computational time from 10ms to 20ms. Therefore, the consumed energy remains unchanged $100\text{mW} \cdot 20\text{ms} = 2\text{mJ}$. Hence, the only possibility to reduce the energy consumption is to reduce the circuit supply voltage V_{dd} . Of course, reducing the supply voltage necessitates the reduction of the frequency in order to ensure correct operation. Since reducing the supply voltage of a digital circuit, the time required for gate signals to settle is prolonged, which, in turn, increases the circuit delay [A.P. Chandrakasan, 1992 [5]] as shown in the Fig.3.3. The mathematical relation between the operating frequency and the supply voltage is later explained.

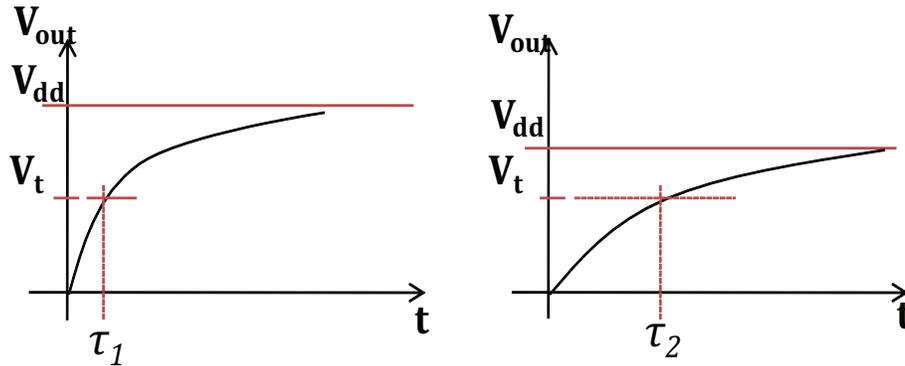


Figure 3.3: Trade-off between supply voltage and charging circuit delay

Dynamic Voltage and Frequency Scaling (DVFS) method enables the elements and is capable to adjust their supply voltage and operating frequency during the application execution. The main objective of this policy is to exploit the energy/delay trade-off to reduce the energy dissipation as illustrated in the Fig.3.4.

As mentioned previously, reducing the supply voltage of a circuit increases the circuit delay, i.e., decreases the operating frequency. Therefore in the Fig.3.4, supposing that t_{nom} is the nominal execution time when the supply voltage and the operating frequency are maximum, and $t_{dvfs} = t_{nom} + t_{slack}$ when applying the DVFS policy. The extension factor between the new and nominal time execution is given as follows:

$$e = \frac{t_{dvfs}}{t_{nom}} = \frac{t_{nom} + t_{slack}}{t_{nom}} = \frac{T_{dvfs}}{T_{nom}} = \frac{f_{nom}}{f_{dvfs}} \quad (3.8)$$

According to [M.T. Schmitz et al., 2004 [3]], this factor allows to lower the supply voltage of the MCUs in accordance to the following formula:

$$V_{dvfs} = V_t + \frac{V_0}{2e} + \sqrt{\left(V_t + \frac{V_0}{2e}\right)^2 - V_t^2} \quad (3.9)$$

where V_t is the threshold voltage, and the constant V_0 is given by:

$$V_0 = \frac{(V_{max} - V_t)^2}{V_{max}} \quad (3.10)$$

Where V_{max} is nominal supply voltage. Based on the energy equation 3.6 and the equation of voltage computation when applying DVFS policy 3.9, the normalized energy is calculated as follows:

$$E_{dvfs} = \frac{E_{max}}{V_{max}^2} \cdot \left(V_t + \frac{V_0}{2e} + \sqrt{\left(V_t + \frac{V_0}{2e}\right)^2 - V_t^2} \right) \quad (3.11)$$

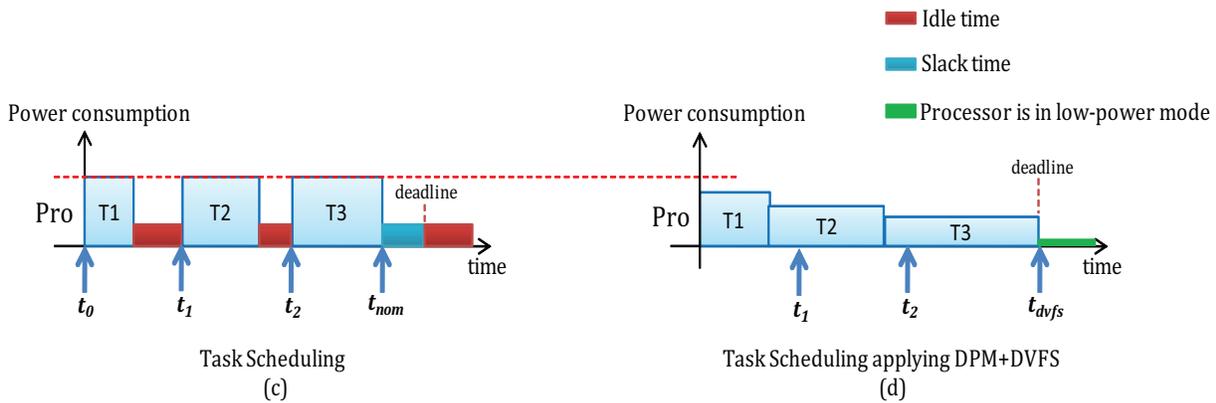


Figure 3.4: Example of using DVFS in the scheduling process

Several strategies can be applied for the use of DVFS method as follows:

- All the tasks are supplied by the same voltage to reduce the energy dissipation.

- Each task is supplied with a different voltage based on the rate of its activity.
- The first task is run with the highest frequency, i.e. with the highest voltage. Based on the actual execution time of this task, the voltage and frequency are regulated consistently for the next task, and so on as described in [M.K. Bhatti et al., 2010 [31]].

3.3 Reconfiguration

Dynamic reconfiguration process is a response to faults during exploitation phase. This process not only consists in reorganizing the material structure of sensor node circuit or network structure, but also includes their control part, in order to allow them to go on with their operation after fault occurrence. From practical point of view, dynamic reconfiguration requires first to detect and localize the faulty part of the node system or the faulty node of network, to analyze the impact on the rest of the system or the network, then to decide a new organization of the system or the network, and to implement corrective actions to reach appropriate configuration.

Among the various objectives that can be assigned to dynamic configuration [Berruet P., 2006 [56]], the following ones can be highlighted:

- to maintain a certain continuity of services whatever the quality of services of the process and control devices is, for example when a hardware failure is encountered on a component of sensor node device that decreases its functionalities, its performances or even its availability; in this case, reconfiguration can be seen as an efficient mean to ensure active dependability of the sensor node system.
- to implement modified control rules, to launch new control services, or to modify control architecture in order to answer to control specifications as required for sensor node operation.

Then implementing such a dynamic reconfiguration concept requires adding a reconfiguration loop to the control systems. This section presents a general framework for on-line reconfiguration involving four major informational, decisional and operational steps such as detection and localization, context analysis, decision making and decision implementation as depicted in the Fig.3.5. The role of these steps is described in the remainder of the section.

3.3.1 Detection and localization

At node level, the self-diagnosis including detection and localization, may be executed as follows: when a hardware failure occurs in sensor node system that creates an abnormal behavior, it will be automatically detected by the Power and Availability Manager (PAM). According to the detection result, the PAM runs localization process to identify correctly the failed component, in order that node system could select appropriate configuration. The detailed description of this self-diagnosis is explained later in the next chapter.

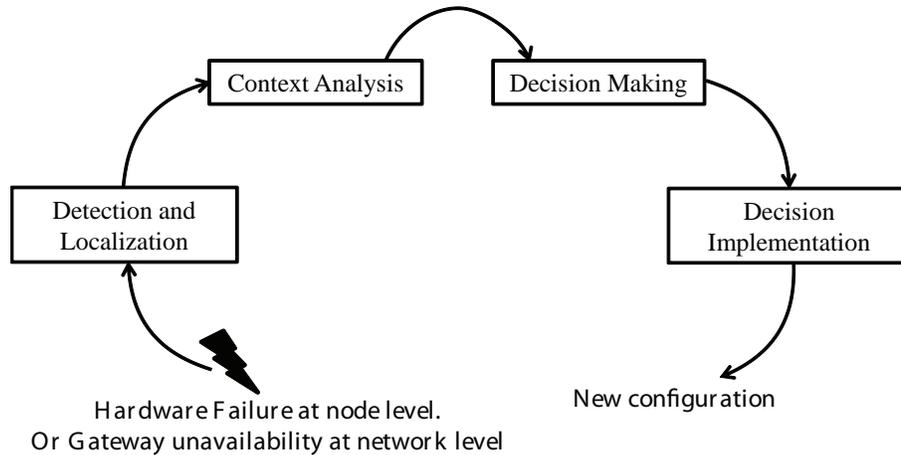


Figure 3.5: Reconfiguration process diagram

At network or cluster level, the Gateway unavailability issue is detected as follows, when the Gateway detects its remaining energy lesser than an energy threshold ($E_{Gateway} < E_{lowerBound}$) that does not guarantee its cluster-head role, the Gateway sends notification to a temporary cluster-head (Tempo-CH) node in its cluster. To understand the role of the Tempo-CH node, you can view the cluster structure that is presented later in the chapter 5. Failed Gateway due to hardware failure is also detected when the Tempo-CH node sends periodically a beacon message to Gateway, if it does not receive any feedback from Gateway, Gateway is considered as failed.

3.3.2 Context analysis

At node level, after completing the detection and localization of the reason of issue, the sensor node runs context analysis. On one hand, this step of reconfiguration process provides context of internal environment of the system as the operating state of node component, or residual energy of node. On the other hand, it also provides the context of external environment of the system. For example, the recharging time of the battery is estimated in consulting the external environment of node.

At cluster level, after the Gateway unavailability issue is detected, the cluster-head selection will be launched by the Tempo-CH. The first step of selection process is the context analysis that is run by the Tempo-CH node. In context analysis, Tempo-CH node sends the request of operating energy level (OEL) value to all cluster-head candidate nodes in its cluster. The computation of this OEL value is explained later in the chapter 5. The Tempo-CH node uses these received OEL values for making decision of new cluster-head selection.

3.3.3 Decision making

At node level, when the context analysis step is completed, sensor node must take its final decision. According to requirement and analysis of the context analysis, this step selects the most consistent configuration to apply. For example, node system selects consistent mode of operation that corresponds to the combination of working components of node in case of energy depletion, or using material support in the case of hardware failure.

At cluster level, after receiving the OEL values from all cluster-head candidate nodes in its cluster, the Tempo-CH node must select the candidate node having the highest OEL value to become new cluster-head of its cluster.

3.3.4 Decision implementation

At node level, this final step implements the new control policy that manages switching from the old configuration to the new one while preserving control continuity and consistency.

At cluster level, the Tempo-CH implements its final decision by sending selection result to all the nodes in its cluster in order to update their routing table.

3.4 Modeling Tools of Discrete Event Systems (DES)

Since its system behavior or system state evolves in function of arrived discrete events such as data sensing, reception and transmission of data packet, hardware failure, energy depletion, etc., therefore wireless sensor node relates to discrete event system (DES). This section is started with a brief introduction of discrete event systems corresponding to the node system that we focus in our study.

3.4.1 DES introduction

Many types of system can be specified by discrete event models including transportation systems, communication systems, production systems, [Y.C. HO, 1989 [7]], or wireless node systems. These are the examples of dynamic systems in which their activities are conducted by the event occurrence at a time instant as the end of task execution, arrival of a product, etc.

Definition

Definition 3.4.1. *The discrete event systems are the dynamic system that evolve according to events observed at discrete points in time [Ramadge and Wonham, 1989 [8]], these systems have two main properties [Cassandras, 1993 [9]]:*

State space is described by a discrete set.

The transitions between the states are realized based on the discrete event occurrences. Several

occurrences of discrete events may be identified as a specific action taken (somebody presses a button), the others may be viewed as a spontaneous occurrence (a computer goes down for whatever reason to be complicated to figure out), while the remaining occurrences may be the result of several conditions which are suddenly all met (the water level in the tank has just exceeded a given value) [Lafortune and Cassandras, 2008 [10]].

The discrete-event systems are divided into: *Time-Driven* and *Event-Driven* systems. To distinguish the difference between two types, let us assume that a clock is existing through which we will measure time, and consider two possibilities:

1. At every clock tick, an event e is to be selected from the event set E . A *null event* is considered when no event takes place, i.e., there is no state change.
2. At various time instants, some event announces that it is occurring.

Thus, the main difference between the two previous assumptions is :

- In the first case, state transitions are synchronized by the clock. At clock tick, the state changes when an event is selected. Therefore, the clock alone is responsible for any possible state transition.
- In the second one, the occurrence of an event, which is not depended in clock tick, creates the state transition. Thus state transitions are the result of combining these asynchronous and concurrent event process.

Hence, states transitions in a discrete event system may whether be synchronized by a clock or be asynchronous with the clock. Obviously, the event-driven systems are more complicated to model and analyze, since there are several asynchronous event-timing mechanisms to be specified as part of our understanding of the system.

Example

Let us consider an example of event-driven DES. A warehouse system as illustrated in Figure 3.6, is considered as a discrete state system, which contains products manufactured in a factory. Whenever a new product is complete at the manufacturing process, it arrives at the warehouse and is stored there. A truck arrives at any time and loads up a certain number of products for delivering them to the clients.

The input $u_1(t)$ and the output $u_2(t)$ describe the number of arriving and departing products, respectively. We define $x(t)$ to be the number of products at time t , and define the output equation for our system to be $y(t) = x(t)$. Since the products are discrete entities, the state space of this system is a set of non-negative integers $1, 2, 3, \dots$. Suppose we choose as input two functions of time defined as follows:

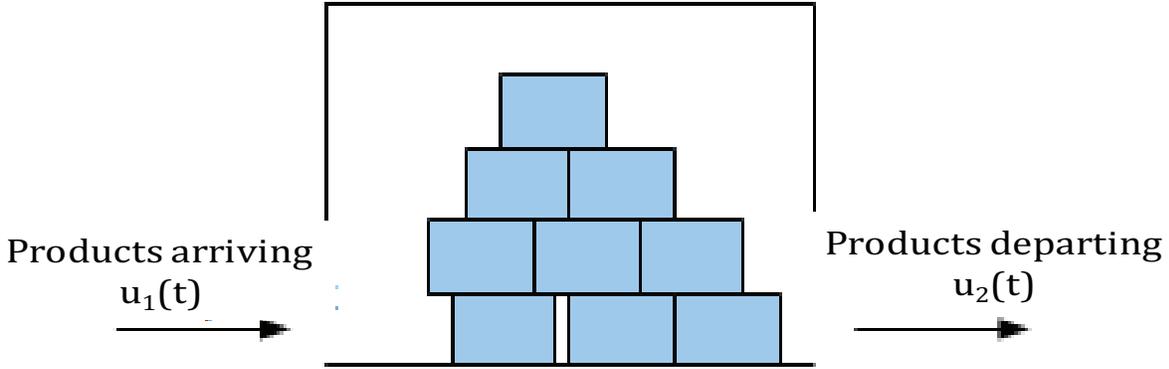


Figure 3.6: Warehouse system

$$u_1(t) = \begin{cases} 1 & \text{if a product arrives at the time } t \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

$$u_2(t) = \begin{cases} 1 & \text{if a product departs at the time } t \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

To simplify the system, let us assume that:

- The warehouse is very large and its storage capacity is never reached.
- Time for loading a product into the truck is zero.
- The truck can only deliver one product at the time t .
- A product arrival and a truck arrival never take place at the same time.

With this assumption, the state equation of the warehouse system is given as follows:

$$x(t^+) = \begin{cases} x(t) + 1 & \text{if } (u_1(t) = 1 \text{ and } u_2(t) = 0) \\ x(t) - 1 & \text{if } (u_1(t) = 0 \text{ and } u_2(t) = 1 \text{ and } x(t) > 0) \\ x(t) & \text{otherwise} \end{cases} \quad (3.14)$$

A typical simple path of this system is depicted in Fig.3.7.

When the sequence of entry events of DES is specified in a deterministic way (i.e. following to the event occurrence, the following system state is unique) as the event sequence $u_1(t), u_2(t)$ without information about the dates of event occurrences, it is possible to obtain a model with logical behavior. If the dates of event occurrences are associated to the events of the sequence, a timed model of the system can be built. And if the information on probability distribution of the events is taken into account, stochastic models are suitable to be used. These three levels of abstraction should be considered at the DES modeling based on the available information,

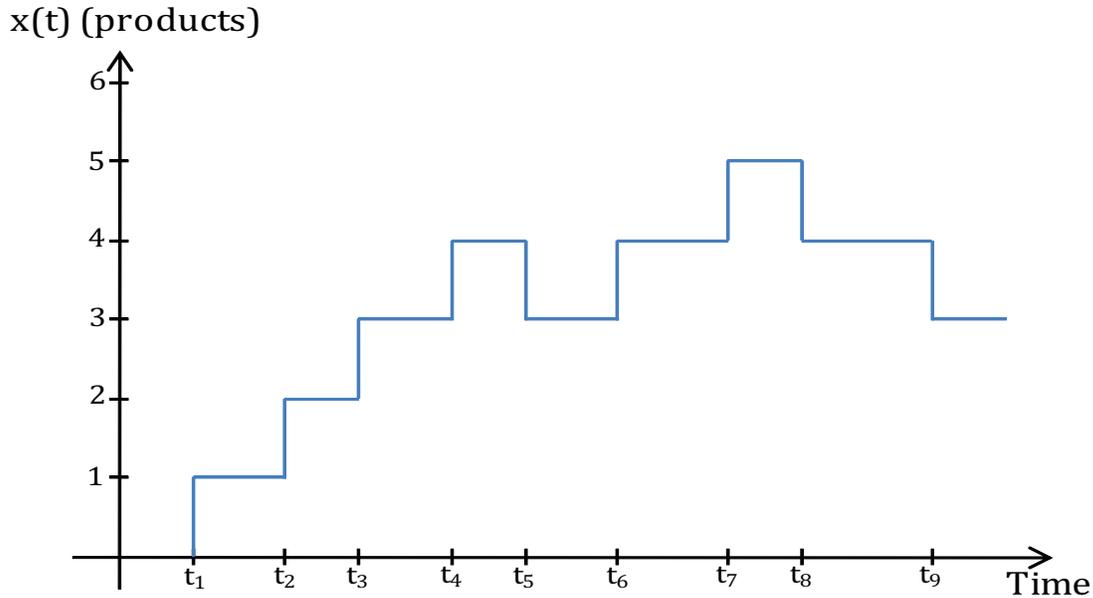


Figure 3.7: Sample path of the warehouse system

and the objective of modeling and analysis of the system. The next section introduces principal tools that allow modeling the logic behavior of discrete event systems, and provides a brief presentation of the available features of each modeling tool.

3.4.2 Tools for DES modeling

Finite State Automaton

This method used to study the logic behavior of a DES based on the theory of language and the finite state automata. The principle of this method is that each DES is associated with a set of events E , which may be considered as an alphabet of a language and the sequences of the events are the words of this language [Lafortune and Cassandras, 2008 [10]]. The finite state automaton is a device that generates a language using this alphabet E following to well-defined rules. That leads to specify the set of the states and the transitions between these states.

Definition 3.4.2. A finite state automaton is defined by 5-uplets: $A = (Q, E, f, q_0, Q_m)$, where Q is the set of finite states.

E is a finite alphabet.

f is transition function, $f: Q \times E \rightarrow Q$

q_0 is initial state, $q_0 \in Q$

Q_m is the set of final state, $Q_m \subseteq Q$

The finite state automaton can be graphically represented by a diagram of state transitions.

It relates to an oriented graph where the states are depicted by the rounds, and the transitions are represented by the arcs between the states. An arc labeled by an event e and connected to two states labeled q and q' , represents a transition of actual state q to next state q' , following to the event occurrence.

The Fig.3.8 describes the graphical representation of a automaton $A = (Q, E, f, q_0, Q_m)$. The set of state of this automaton is $Q = (q_0, q_1, q_2)$. The initial state is indicated by an entry arrow. The final states are represented by the double rounds, $Q_m = (q_0, q_2)$. The set of symbols (labels) associated with the arcs of diagram is the event set $E = (e_0, e_1, e_2)$. The arcs of diagram are the graphical representation of the transitions between the states of the finite state automaton, $f = Q \times E \rightarrow Q$:

$$\begin{aligned} f(q_0, e_0) = q_0, f(q_0, e_2) = q_2, f(q_1, e_0) = q_0, f(q_1, e_1) = q_1, f(q_2, e_1) = q_2, \\ f(q_2, e_0) = f(q_2, e_2) = q_1 \end{aligned}$$

The notation $q_0, e_2=q_2$ means that if the automaton is in state q_0 , at the occurrence of the event e_2 , the automaton evolves to state q_2 . Two important remarks are given from this example:

1. An event may occur and do not change the automaton state (i.e. $f(q_0, e_0) = q_0$)
2. For a given state, two different events may generate the transitions to similar state $f(q_2, e_0) = f(q_2, e_2) = q_1$

An automaton is determinist if its initial state is unique and its relation of transition, applied to a couple q_i, e_i always defines a unique state. The automata that do not represent this property are said non-determinist. The biggest drawback of this modeling type is the combinatorial explosion of number of the states when it is used to represent the complex systems. To overcome this limitation, the state machines was extended that leads to appearance of statechart, which is then used to be integrated in UML (Unified Modeling Language) to model oriented objects. The Petri Net is another remarkable modeling tool in DES domain that deals with the state explosion issue.

StateChart

The StateCharts are developed by David Harel in 1983 as formalism for the specification of reactive behavior [D. Harel, 1988 [11]]. The StateCharts are the extended FSMs (Finite State Machines) with the concepts of hierarchy and parallelism that allows representing a system in more compact way, in order to avoid the combinatorial explosion.

The StateCharts describe the system behavior, i.e. how the components of the system communicate between them, cooperate and how these components realize their internal proper behaviors. The behavior of the system is described through all possible states of its components,

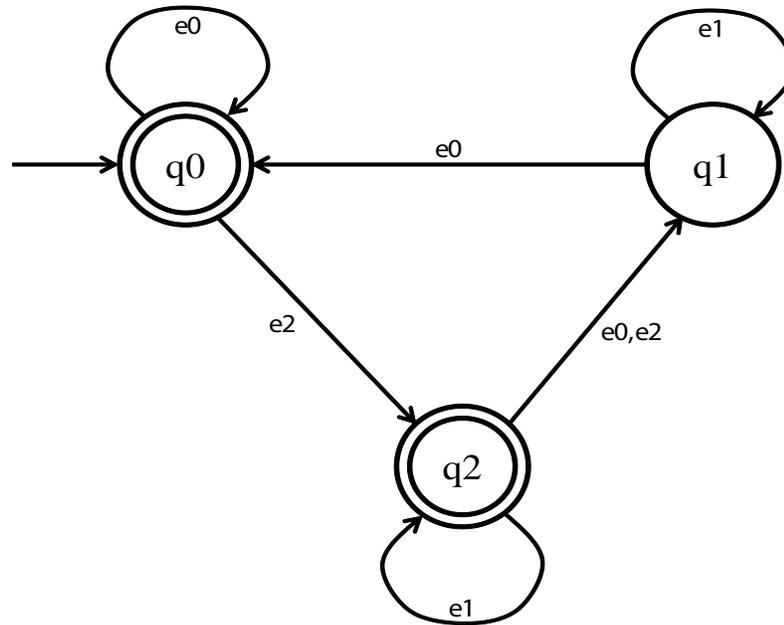


Figure 3.8: Diagram of state transitions

and in the manner in which the events act on these components. A State Diagram is composed of:

- **States:** they are represented by the rounded boxes, the states can be terminal or abstract. The initial state indicated by the end of arc whose origin is not a state but a small circle. An abstract state may show several separated areas by dotted lines. In this case, each area contains a description of a state diagram and these state diagrams evolve in parallel. The synchronizations between these parallel states are guaranteed by events whose distribution is instantaneous throughout the state diagram;
- **Transitions:** they are represented by the oriented arcs that connect two states.
- **Trigger events:** they can be external or internal to the state diagram and cause transitions between states. They are shown as labels on transitions.
- **Conditions:** they are Boolean expressions that describe the events. These conditions are represented in the transitions between the brackets as follows: event [condition]
- **Generated events or actions:** they are broadcast at the crossing of an arc. They are represented on the transitions with the event and the condition just after the symbol "/". The general form of the label of an arc in a state diagram is $e_i[c_j]e_0$ with e_i trigger event, c_j condition, and e_0 generated event generated.

The Fig.3.9 gives an example of the graphical representation of a statechart. In this diagram the state s_0 is divided into a three automaton states s_1, s_2, s_3 . The state s_1 is itself splitted into

two states s_4 and s_5 , which in turn are decomposed respectively s_6, s_7 , and s_8, s_9 . The states s_0, s_1, s_4, s_5 are called composite states. The transitions are labeled by *events/actions* as r_1/a_1 .

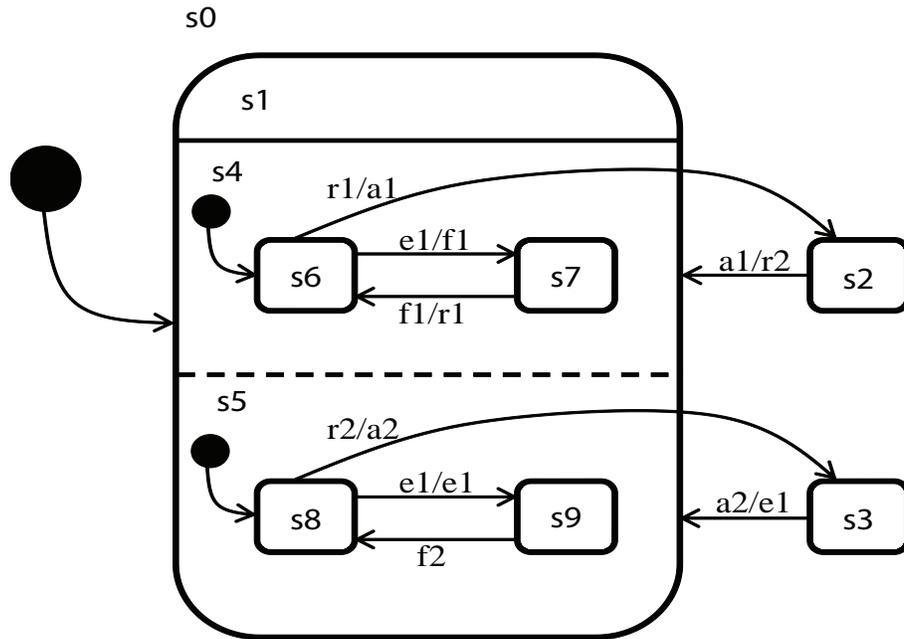


Figure 3.9: Diagram of statechart

The state of the system is modeled by the configurations, i.e. by the state sets. For example in the Fig.3.9, the system can be in following configurations: (s_1, s_6, s_8) , (s_1, s_6, s_9) , (s_1, s_7, s_8) , (s_1, s_7, s_9) , (s_2) , (s_3) . A transition is only triggered if its original state belongs to current state of the system. For example, if the system is in the configuration (s_1, s_6, s_9) , the transition labeled by r_1/a_1 may be triggered, then the original state will be left and the state (s_2) will be reached.

Petri Net

The Petri Nets were developed by German mathematician Carl Adam Petri in 1962. At beginning, this tool allows describing the existing relations between conditions and the events. Thereafter, the Petri Nets have been enriched on all aspects related to the modeling of parallel and distributed behaviors [J.L. Peterson, 1981 [12]].

Graphically in Petri Net, the events are associated with the transitions(T:), and the places(P:) contains the conditions, from which the transitions can take place. Some such places are viewed as the input to transitions. Other places are viewed as the output of transitions, they are associated with conditions that are affected by the occurrence of these transitions.

A transition is connected to its input places by input arcs shown as directional arrows. Conversely, output arcs drawn from the transitions to its output places. Places may contain zero,

one, or more tokens that indicate the holding conditions. The distribution of tokens in places refers to the marking of PN. Arcs may have different multiplicities. A transition is enabled if its input places have a number of tokens equal or greater than the multiplicity of the input arcs. The firing of a transition removes the tokens from its input places and deposits the tokens in its output places.

Definition 3.4.3. A Petri Net [Lafortune and Cassandras, 2008 [10]] may be defined by 4-uplets $R = (P, T, Pre, Post)$, where:

P is the finite set of places

T is the finite set of transitions

$Pre: P \times T \rightarrow N$ is the set of previous places that define the arcs from the places to the transitions.

$Post: P \times T \rightarrow N$ is the set of following places that define the arcs from the transitions to the places.

$Pre(p,t)$ is the weight of the arc from the place p to the transition t ; $Post(p,t)$ is the weight of the arc from the transition t to the place p .

A marked Petri Net is defined as a couple (R,m) in which R is a Petri Net and $m: P \rightarrow N$ is an application called marking. In Petri Net, a transition is sensible or enabled if:

$$\forall p \in P \quad M(p) \geq Pre(p,t)$$

where $M(p)$ represents the number of tokens contained in the place p . The evolution of the marking of a place p at the firing of a transition t is computed by following equation:

$$M'(p) = M(p) + C(p,t)$$

where C is the incidence function defined by:

$$P \times T \xrightarrow{C} N$$

$$C(p,t) = Post(p,t) - Pre(p,t)$$

and represents the variation of marking induced on the place p by the firing of the transition t .

In Petri Net modeling, the events are associated with the transitions(T:), and the places(P:) contain the conditions, from which the transitions can take place. Places may contain zero, one, or more tokens that indicate the holding conditions. The connection between the places and the transitions is drawn by the arcs. A transition is enabled if its input places have a number of tokens equal or greater than the multiplicity of the input arcs. The firing of a transition removes the tokens from its input places and deposits the tokens in its output places. Considering the Petri Net as described in Fig.3.10. This Petri Net example is formulated by the set of places $P = (p_1, p_2, p_3, p_4, p_5)$. The set of transitions is $T = (t_1, t_2, t_3, t_4)$. We assume that the actual marking is $M_2 = [01100]^T$. The i^{nd} component of the column vector (for simplicity we write this vector in its transposed form) is the number of tokens associated with the place p_i . In this

actual marking, the places p_2 and p_3 contain a token in each. Two transitions are enabled by the marking $M_2 : t_2$ and t_3 . Thus, the next evolution of this Petri Net may correspond to the firing of the transition t_2 or the transition t_3 , but the firings of other transitions are not possible from this marking. The firing of the transition t_3 conducting to the marking $M_2 = [01001]^T$, when the transition t_2 conducting to the marking $M_4 = [00110]^T$. For the firing of the transition t_4 , necessary marking is $M_5 = [00011]^T$ that corresponding to the previous firings of the transition t_2 and t_3 . After the firing of the transition t_4 , the system is back to its initial marking $M_0 = [10000]^T$.

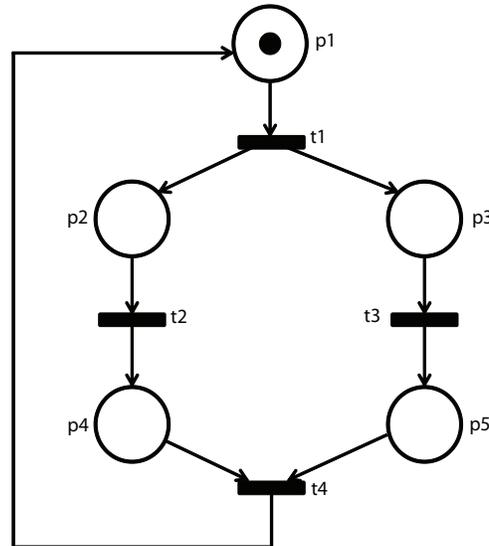


Figure 3.10: Marked Petri Net

The Petri Nets allow modeling different situations presented in the system as sharing of the resources, synchronization of the activities, concurrent or parallel activities, etc. It exists several Petri Net families relative to data enrichment (colored PN, predicate-transition PN, object PN, etc.), the other Petri Net families relative to time enrichment (timed PN), and the others are related to stochastic PN. Based on these various type of Petri Net families and the ability of combination of these families together, Petri Net is a powerful tool to model complex, dynamic, and discrete systems as wireless node system in our case study.

Since considered wireless node relates to discrete event system where its operating state evolves according to unpredictable events observed at discrete points in time (e.g. hardware failures), discrete Time Stochastic Petri Net (TSPN) modeling [Molloy M. K., 1985 [25]] is suitable to model sensor node operation. The TSPN model not only performs the concurrent operations and asynchronous events of node system, but also supports the failure prediction features. The timed and stochastic features of TSPN for node system modeling are depicted in our study as following:

- The immediate transitions are modelled as black bars.
- The timed transitions are modelled as white rectangles.
- The firing rate λ is assigned to each arc to depict the firing probability of each transition. By default, arcs without assigning rates represent the firing probability of 1 for corresponding transitions.

An example of TSPN describing the data capture for being processed in node system is depicted in Fig.3.11, which includes modeling of failure occurrence and modeling of failure detection. While the first modeling contains the failure generator of sensing component for data capture, the second modeling includes the data storage process using timed transition attached to watchdog timer, of which the deadline is T . As illustrated in this figure, the place $P:UpS$ represents number of available sensing component and a token appears in it at beginning. When the data capture is asked, the sensing device begins capture process corresponding to the firing of the transition $T:BeginCapture$, which causes the token distribution in the two places $P:WaitForCapCompletion$ and $P:WaitForData$. While the first place represents the waiting state of data capture completion, the second place depicts the waiting of data appearance in capturing Fifo. The failure of sensing device is represented by the transition $T:NotCap$ with rate λ , while the transition $T:Cap$ represents the successful data capture with rate $1 - \lambda$. On one hand, the firing of the transition $T:Cap$ distributes a token in the place $P:ReadyData$ that leads to the firing of the transition $T:EndCapture$. This transition firing causes the token redistribution in the place $P:UpS$. On the other hand, the firing of the transition $T:NotCap$ leads to the firing of the transition $T:FailS$ when watchdog timer is over, which distributes a token in the place $P:DownS$ that represents the failure of sensing component. This modeling principle is reused for simulation model of wireless node behavior to compute node availability in the next chapter.

3.5 Conclusion

We have introduced in this section the basic background for reducing energy consumption and improving availability. Throughout this chapter, we gave an overview of two major constraints and their impacts. Then the techniques for energy reduction are also presented. Besides, the way in which the node system evolves and runs reconfiguration in case of failure is also discussed. The DES modeling tools that allow describing node functionality are then presented through brief description. The next chapter will describe in detail our novel design approach including energy-efficiency solution, self-diagnosis of hardware detection and localization, material reconfiguration, and our evaluation through simulation and implementation results.

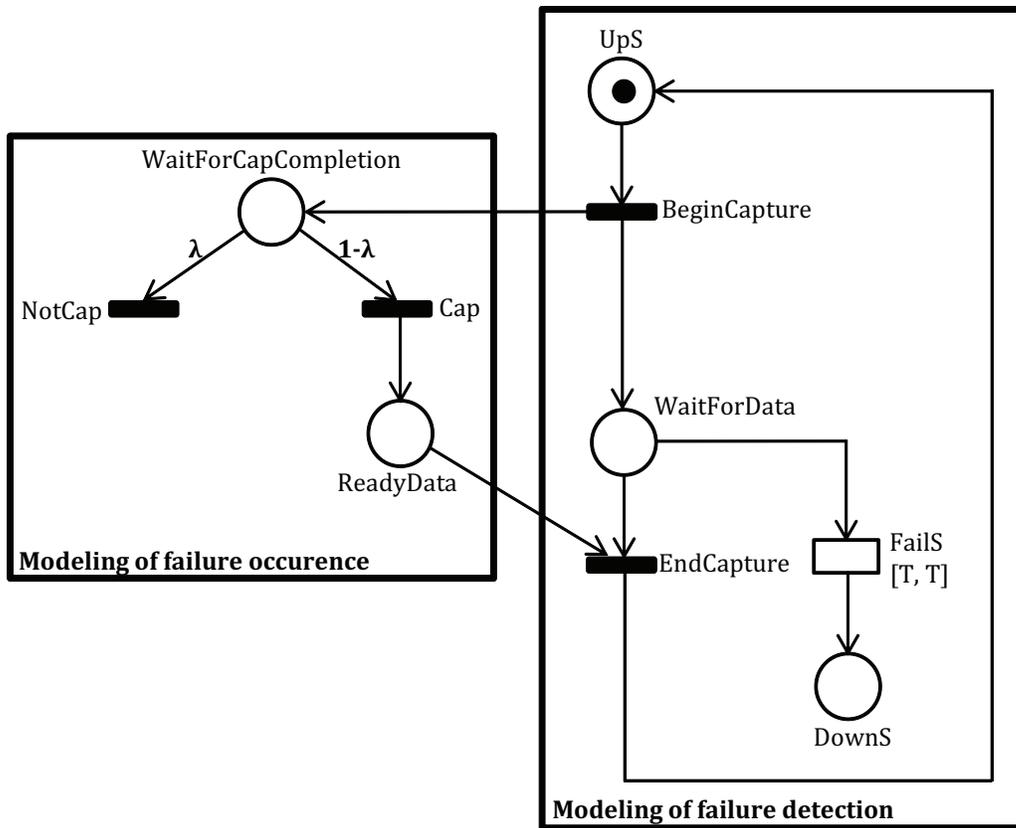


Figure 3.11: Timed Stochastic Petri Net Modeling

Chapter 4

APPROACH AT NODE LEVEL

We live in a world filled with sensors. The buildings that we work and live in have sensor monitoring temperature, smoke and fire, security. The cars, the roads and the parking lots [Streetline project, 2014 [70]] are equipped with cheap and low-power sensor nodes to help drivers to find unoccupied parking places and avoid traffic jams. This solution can significantly improve the city traffic and reduce the emission of carbon dioxide. Manufacturing environments need sensors because you cannot control what you cannot measure. Making products, while meeting safety, quality and efficiency targets, requires a lot of sensors.

Many breakthroughs in the MEMS revolution make sensor node become much smaller, less expensive and low power in the last few decades. However, sensor technology is also driven by Moore's law that states that the number of transistors in integrated circuits doubles approximately every two years [G.E. Moore, 2006 [6]]. This law has proven to be correct by the technological advancement. The increase in transistor number leads to the increase in computational power that impacts the power utilization of these complex devices. Since sensor nodes are powered by portable batteries, they have a limited energy budget. Therefore, this increase in power consumption plus ineffective use of battery energy will highly shorten the sensor node lifetime. As a result, that reduces WSN lifetime and reliability.

On the other hand, when deploying in harsh environments such as Tsunami Detection [K. Casey, 2008 [36]], Hazardous Gas or Intrusive Detection [Nicolas Ferry, 2011 [26]], Glacial Environment Monitoring [K. Martinez, 2005 [40]], hardware failure faults inside sensor node is unavoidable and unpredictable that may lead to crash the sensor node. This is a critical issue because if the crashing node is not promptly detected and repaired, its consequences can be severe in term of human life, environment impact or economic loss. That stresses the need of self-diagnosis method to promptly determine whether a node component is still operating or not, in order to react consistently and make the node less vulnerable.

This chapter introduces firstly the novel design approach of our smart sensor node, and then all its potential issues related to energy and hardware failures in Section 4.1.1. Section 4.2 describes in detail our approach to reduce energy dissipation and improve availability for

wireless sensor node. Then our approach is evaluated through simulation results in Section 4.3 and hardware implementation in Section 4.4. Finally, several conclusions are given in Section 6.3.1.

4.1 Introduction

As explained above, wireless sensor node is powered by battery, hence its autonomy depends on the battery life. The more energy is consumed, the more the node lifetime is reduced. Therefore, minimizing the energy consumption is a big challenge for the designers. Additionally, when the sensor nodes are deployed in the harsh environment in several applications, each node is a failure-prone of the component, hence the failure detection and localization is necessary for sensor node to react against these problems. That allows sensor node to make promptly reconfiguration decision to keep sensor node still operating. The next sub-section introduces our novel hardware design for wireless sensor node that aims to deal with these issues.

4.1.1 Novel hardware design of wireless sensor node

Next to the components existing in the sensor node as described in Section 1.1, our node includes the Power and Availability Manager (PAM) block combined with FPGA as illustrated in Fig.4.1, in which:

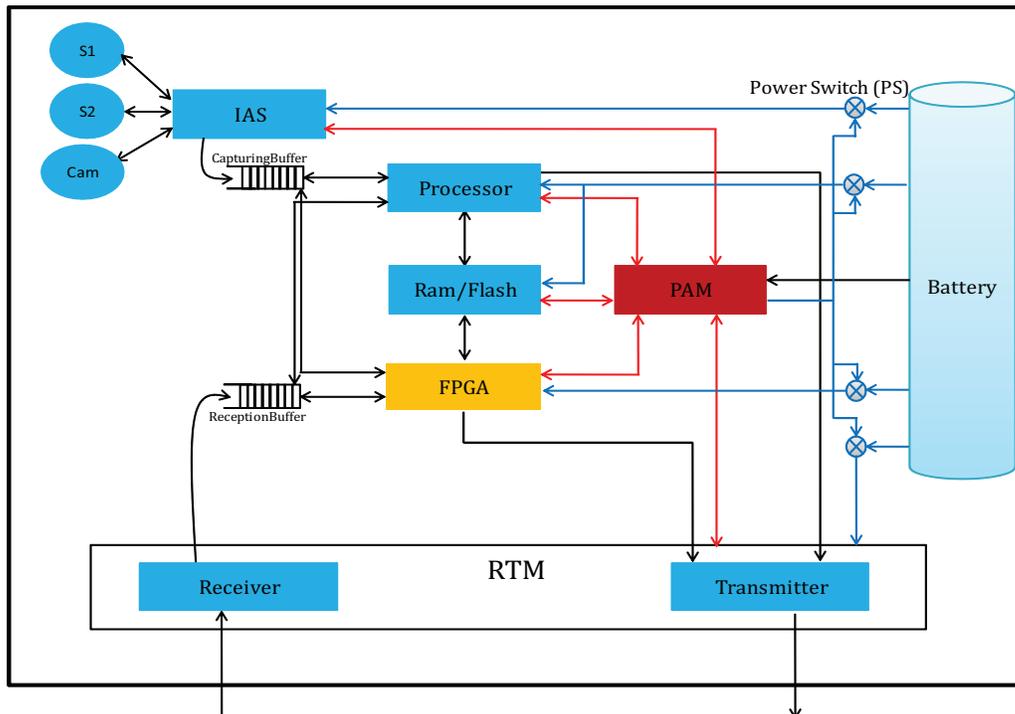


Figure 4.1: Block diagram of wireless sensor node

- *Power Availability Manager (PAM)* is the intelligent part of our sensor node that controls efficiently the energy consumption of the node system. Furthermore, our PAM block also supports the node to detect the failing component, and then take an appropriate corrective solution to make node system less vulnerable.
- *Configurable zone of FPGA* is included in sensor node to increase the availability of sensor node when the processor or the Ram memory is down. Moreover, it can be used to enhance the node performance for signal and video processing.

To mitigate the data conflict, two First In First Out (FIFO) buffers are used in which CapturingBuffer stores the captured data, and ReceptionBuffer saves the data sent by other nodes. Next section presents the potential issues that may occur inside sensor node.

4.1.2 Issues and corrective solutions

Based on the list of possible issues [M. Ringwald, 2007 [54]], and our knowledge enhanced by Eryma's professional experience, several problems encountered in a sensor node and their corrective solutions are described in Table 4.1.

Table 4.1: Issues and corrective solutions for sensor node

Problem	Software and Hardware causes					Energy causes		Solution
	Down Processor	Down Ram	Down IAS	Down RTM	Soft Bug	Low Energy	Energy Depletion	
Dead Node	X							PAM enables extra processor to replace it
		X						PAM enables extra memory to replace it
							X	Wait for recharging battery
Malfunctioning Node			X					PAM changes mode of operation to relay point
				X				PAM changes mode of operation to local processing
					X			Processor reboots
						X		PAM selects suitable operating mode and waits for battery recharge

According to Table 4.1, with the help of PAM and FPGA block, our self-reconfigurable sensor node can detect wrong behaviors and failures due to software, hardware or energy when they occur, then take corrective solution by choosing an appropriate operating state or running reconfiguration, in order to make itself less vulnerable. Thus, it can react consistently against all issues. Besides, PAM block not only intervenes in case of failure, but also selects the suitable mode of operation to minimize the power consumption, that leads to extend the node lifetime.

4.2 Approach for Energy-Efficiency and Availability Improvement

4.2.1 Energy management

In this section, we describe how to implement both Dynamic Power Management and Dynamic Power and Frequency Scaling methods to reduce energy consumption in wireless sensor node. As described above, Dynamic Power Management (DPM) is a well-known technique that selectively shutdowns or puts to sleep some active components in embedded system of sensor node, to manage energy consumption and heat dissipation problems. This technique provides different low-power modes:

- Idle mode in which the computational core is shut down while leaving its peripherals active.
- Sleep mode in which the sensing, communicating components, the computational core and its peripherals that require the system clock are shut down, leaving the peripherals that use their own clock, or the clock from other devices, active.
- Deep Sleep mode in which the sensing, communicating components, the computational core and its peripherals (except Real-Time Clock RTCC and Deep Sleep Watchdog Timer), Flash memory are shut down.

Meanwhile, Dynamic Voltage and Frequency Scaling (DVFS) technique is widely used for power management in modern processors, as it is an effective method for achieving low power consumption while meeting the performance requirements. The purpose of DVFS techniques is to scale dynamically the circuit speed and the supply voltage level of processing and communicating components, to process node system workload. The frequency and supply voltage are directly related to power consumption in CMOS technology as explained in Section 3.2.3. As shown in Section 3.2.3, the use of DVFS method comes at expense of power overhead due to its complex computation. To address this issue, only two values of supply voltage and corresponding operating frequency are selected according to the type of received events (as described in Fig.4.2), in which:

- Node components (processor, memory, radio module) are supplied with low voltage and correspondingly low frequency when received data contains no hazardous event called normal event.
- Otherwise, when a hazardous event arrives, these components are powered with high voltage and operate at correspondingly high frequency.

The effectiveness of DPM and DVFS implementation will be evaluated in sub-section 4.2.3.

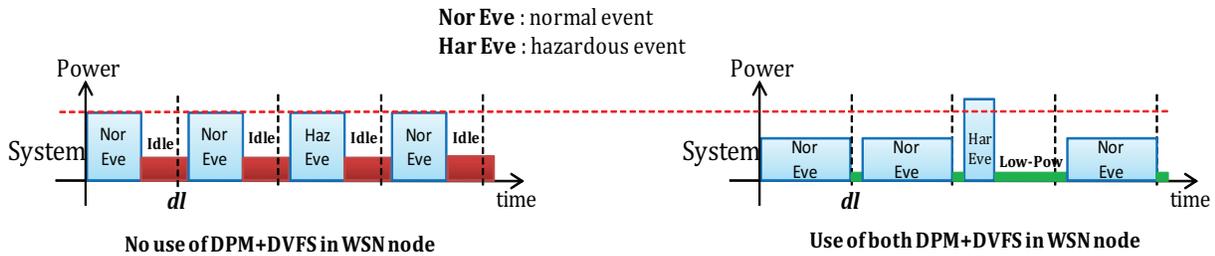


Figure 4.2: Principle of DPM and DVFS implementation in wireless sensor node

4.2.2 Availability management

This sub-section introduces a hybrid self-diagnosis design by combining on-line measurement of consumed current of node subsystem with functioning test of each node component, in order to detect and localize precisely the failing component. Our method deals with hardware faults that lead to avoid software faults [R. Szewczyk, 2004 [42]]. The main contributions of our approach are:

- Detecting hardware failures inside sensor node based on real-time current measurement, then reacting promptly against these hard faults to avoid any serious consequences and to keep sensor node still functioning.
- Localizing accurately the failing component that allows sensor node to select consistent solution.
- Providing a low-power approach because sensor node is battery-powered device that has a limited energy budget.
- Presenting a low-cost solution for the trade-off between sensor node cost and maintenance cost by adding extra components for critical components such as sensing and actuating components, processor, memory, etc. Therefore the failing components are replaced autonomously by their redundancies instead of replacing hard-faulty node by a new one. That also avoids the cost of a human intervention.

The principle of our detection approach of hardware failure is based on exceeding a threshold of consumption measurement. To realize the real-time measurement of the node supply current, an additional device is added. This device cooperates with the Power and Availability Management (PAM) device in our node architecture, in order to detect the failing component, and then take an appropriate solution to make node system less vulnerable. Two material solutions are proposed to realize online current measurement. Fig.4.3 shows the first material implementation, which includes a shunt resistor placed in the output of the circuit, then its voltage $V_{shunt} = I_{out} * R_{shunt}$ is measured and amplified before entering into the ADC input of PAM block.

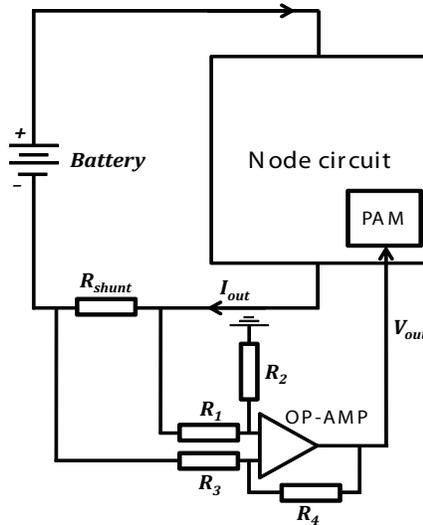


Figure 4.3: Principle of first current measurement implementation

In the second implementation, a current measurement device called HALL probe is used to measure supply current as depicted in Fig.4.4. This device offers small size for space saving, low-power consumption, accurate measurement with wide current measurement range, high immunity to external interference and low-cost.

This device converts measured current into voltage as following equation:

$$V_{out} = V_{ref} + \frac{0.625 * I_P}{I_{ref}} \quad (4.1)$$

In the above equation, V_{ref} is reference voltage, $V_{ref} = 2.5 \pm 0.025V$ by default. While I_{ref} is internal reference current and its value can be changed to have three values as 5A, 10A, 20A. That allows a large current measurement range [-60A, 60A], and $I_{ref} = 5A$ is selected in our work. When a node component encounters a hardware failure, two cases must be considered:

- Low current due to rupture of supply voltage line or rupture in component circuitry. For example, complex circuitry of sensing component makes it very sensible in hardware failure.
- An overvoltage or short-circuit due to stress from environment conditions, inadequate enclosures, or technical shortcomings.

To accurately detect a hardware failure incident inside sensor node, two voltage thresholds are set by the PAM to create a voltage range $[V_{lowerBound}, V_{upperBound}]$. If the measured voltage is outside this range, a hardware failure is detected inside sensor node. The value of these voltage thresholds is dynamically regulated to node operating mode, which is technically described in implementation section. When a hardware failure is detected, we must identify the

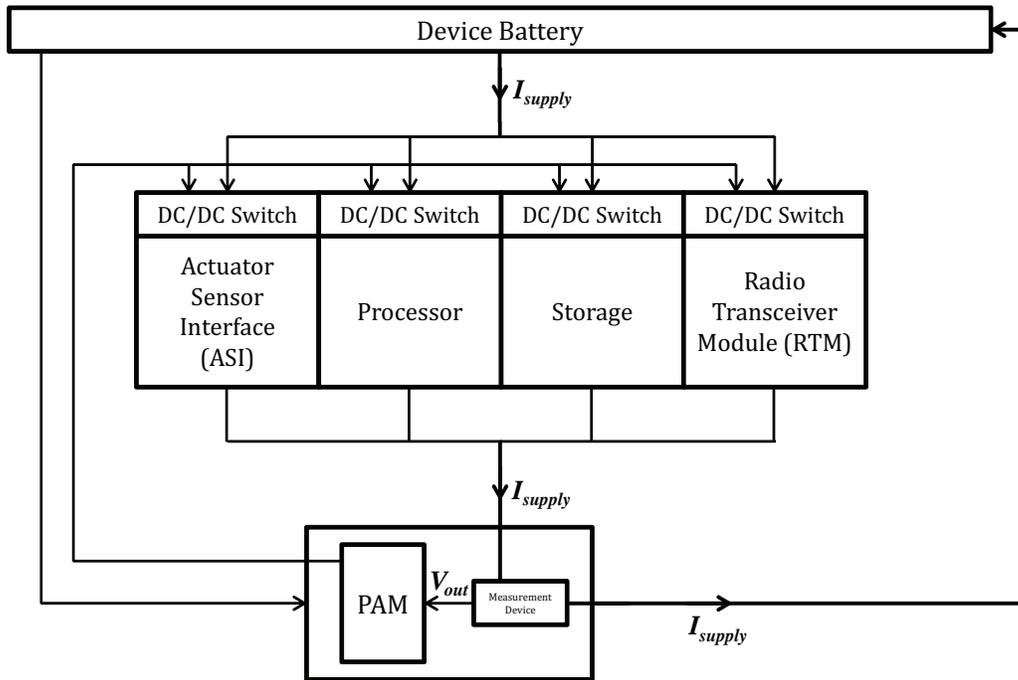


Figure 4.4: Real-time current measurement using HALL probe

failing component. In practice, some gaps between the power consumptions of node components are small (about 100mW [Nicolas Ferry, 2011 [26]]), hence the current measurement is not always able to localize precisely the failing component. Therefore, additional functioning test is required in addition to combine with online current measurement. When hardware failure is detected based on measurement, the PAM stops current operation of sensor node in order to run functioning tests for each node component.

As explained in section 3.4.2, the discrete Time Stochastic Petri Net (TSPN) [Molloy M. K., 1985 [25]] is selected to model wireless node operation because its system relate to discrete event system. Since hardware failures are considered as unpredictable events observed at discrete points in time, thus we use TSPN model to describe the hardware failure localization approach as depicted in Fig.4.5. The timed and stochastic features of TSPN modeling for localization process are depicted in this figure as following:

- The immediate transitions are modelled as black bars.
- The timed transitions are modelled as white rectangles.
- The firing rate λ is assigned to each arc to depict the firing probability of each transition. By default, arcs without assigning rates represent the firing probability of 1 for corresponding transitions.

Also, the inhibitor arc from one place to a transition means that the firing of this transition is prohibited when a token appears in this place. As seen in Fig.4.5, the PAM state is in sleep mode

when no hardware failure is detected by current measurement device, which is represented by the appearance of a token in the place $P:PAMIsSleep$. A model of hardware failure generator is employed as described in black rectangle, where the places $P:UpP$, $P:UpRam$, $P:UpS$, $P:UpRadio$ represent the available node component such as processor, ram memory, sensor and radio transceiver module. At beginning, one token resides in each place as seen in Fig.4.5. The failure rates of these components are respectively depicted by exponentially distributed firing rates λ_p , λ_{ram} , λ_s , λ_{radio} .

When any component is failed that corresponds to the firing of one among four transitions $T:FailP$, $T:FailRam$, $T:FailS$, $T:FailRadio$, an abnormal gap in current measurement is detected and a token appears in the place $P:MeaPowIsAbnormal$ (the current measurement model is not described here). The value of converted voltage may be lower to $V_{lowerBound}$ or higher to $V_{upperBound}$. Then, an interruption signal is generated by one of two analog comparators that makes the PAM waken-up from Sleep mode, which corresponds to the firing of the transition $T:EnablePAM$. A token appears in the place $P:PAMIsWakenUp$. Afterwards, the PAM stops current running operation corresponding to firing of the transition $T:StopAllRunningOps$, which puts a token in the place $P:WaitHardFailLocalStart$. The transition $T:StartHardFailLocalization$ is then fired to enable the self-diagnosis sequentially for each node component (processor, memory, sensor and radio).

The principle of self-diagnosis is to associate each component with a functioning test as follows:

- **Processor:** the PAM sends a message to processor, and waits the feedback. After a time interval with no response, the processor is considered as failed.
- **Ram:** the PAM writes a testing data in Ram, then reads this data and compares it with the original data. If similar, Ram memory is still available, otherwise it is down.
- **Sensor:** the PAM makes a request of capturing data. If no data arrives in capturing buffer during a time interval, the sensor is considered as failed.
- **Radio transceiver (RTM):** the PAM makes a request of data transmission to a neighbor node. At the same time, current consumption of data transmission is measured by a dedicated current measurement device to output a corresponding voltage. If this voltage is larger than given voltage threshold, the radio transceiver is still available, otherwise it is considered as failed.

When self-diagnosis processes of all components are completed corresponding to appearance of one token in the place $P:WaitHardFailLocalEnd$, it leads to the firing of the transition $T:EndHardFailLocalization$ which gets the PAM return to Sleep mode. As seen in Fig.4.5, any failing component is localized after the completion of its self-diagnosis process. As explained previously, the localization process including functioning tests is only executed when a hardware failure is detected by measurement, which allows our approach to economize large amount of energy. In order to avoid harming the circuit of sensor node due to overvoltage, the PAM

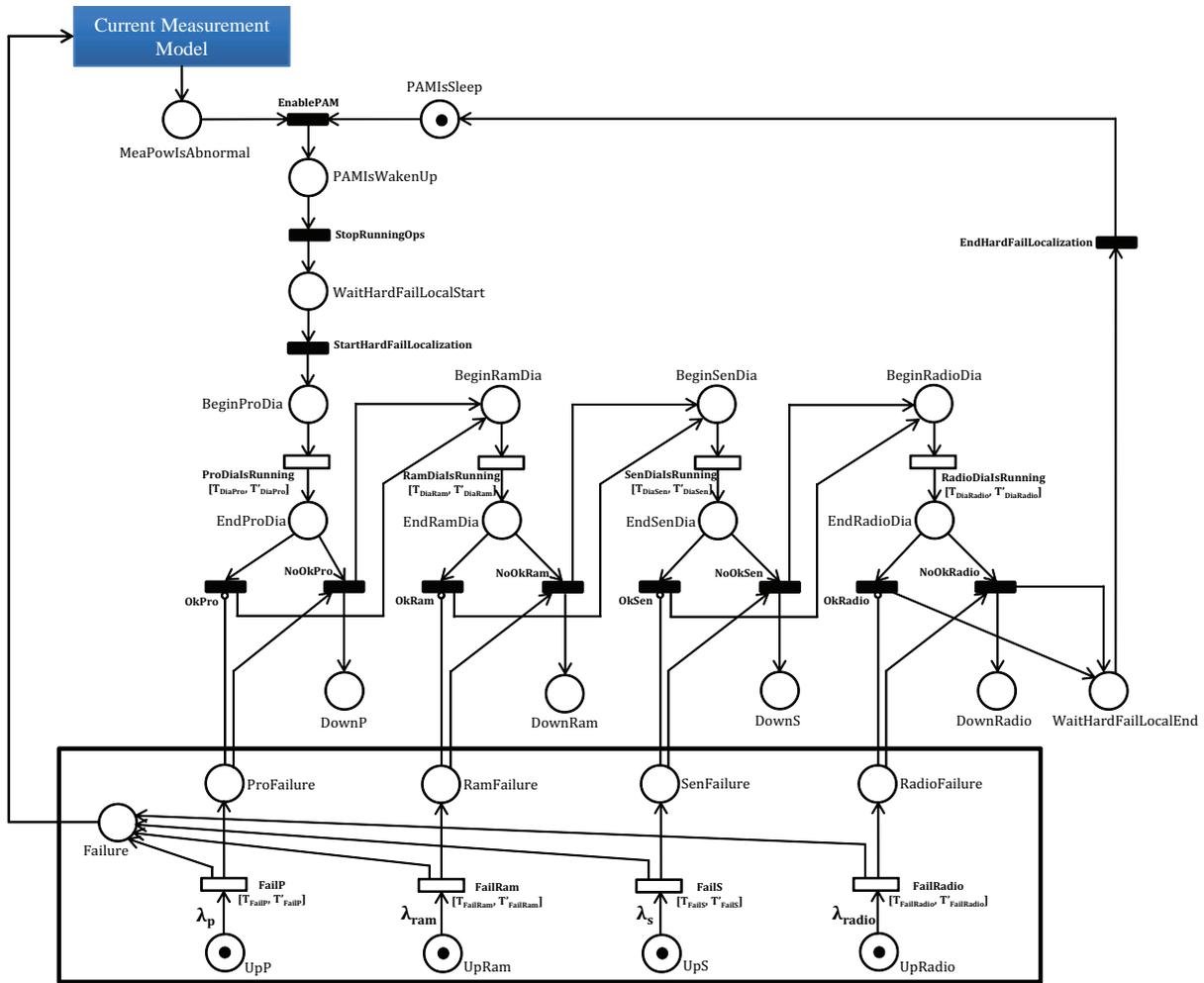


Figure 4.5: Hardware failure localization process

supervises consumed current of each node component during its functioning test. If an overvoltage occurs during functioning test of one component, PAM stops immediately this test and this component is considered as failed. Then the functioning test is continued for other components.

Once a failed component is identified, node system executes the hardware reconfiguration or changes the operating mode as summarized in Tab.4.1, in order to maintain the node operation. Based on all the issues listed in Tab.4.1, we define eleven operating modes for a sensor node that is presented in next sub-section.

4.2.3 Node functionality modeling

Eleven operating modes for a sensor node including their active and inactive components are defined, which are presented in Table 4.2.

Table 4.2: Operating modes of sensor node

Mode \ Unit	Processor	RAM	FPGA	IAS			RTM	
				Sensor1	Sensor2	Camera	Transmitter	Receiver
On-Duty	On	On	Off	On	On/Off	On/Off	On	On
Performance Enhance	On	On	On	On	On/Off	On/Off	On	On
Dead Processor	Off	On	On	On	On/Off	On/Off	On	On
Dead RAM	On	Off	On	On	On/Off	On/Off	On	On
Local Processing	On/Off	On/Off	On/Off	On	On/Off	On/Off	Off	Off
Relay	On/Off	On/Off	On/Off	Off	Off	Off	On	On
Monitoring	On/Off	On/Off	On/Off	On	Off	Off	Off	On
Observation	Sleep	Off	Off	On	Off	Off	Off	On
Sleep	Sleep	Off	Off	Off	Off	Off	Off	Off
Deep Sleep	Deep Sleep	Off	Off	Off	Off	Off	Off	Off
Dead Node	Off	Off	Off	Off	Off	Off	Off	Off

The behavior of node system refers to the Discrete Event System as defined in section sec:ModelingToolsOfDiscreteEventSystemsDES. Since there is no concurrencies, synchronization and the number of operating modes of sensor node is limited, Finite State Machine (FSM) is suitable for modeling these operating modes (see Fig.4.6) among all models. This FSM model consists of a set of states and transitions. When each state represents a particular mode (On-Duty, Performance Enhance, Monitoring, Observation,...), and each transition represents one or more discrete events that make the transition from one operating mode to another one. The FSM model is divided into two parts marked with blue border and green rectangle. The blue one mentions the availability management of the system, while the green one relates to the compromise between performance and energy-efficiency. At beginning, FSM model is initially in the Monitoring state, only Processor, Ram, Sensor1, and Receiver are active.

Because of limited battery, the energy-efficient consumption in sensor node is always a critical issue for designers, even in case of harvesting energy because the energy from the environment is generally unpredictable, discontinuous, and unstable. Additionally, the more energy is saved, the more node lifetime is extended. In our approach, sensor node can harvest energy from the environment by using Weather Forecasts (WFs) [Nicolas Ferry, 2011 [26]] model. WFs are used to determine which power harvesting source will have the highest energy availability and to predict the node lifetime. The energy management of our node is controlled by PAM block based on both DPM and DVFS techniques as explained previously. In the Fig.4.6, the states are arranged in order of increasingly consuming energy such as Deep Sleep, Sleep, Observation, and Monitoring. Based on the activity period and the battery level that are represented

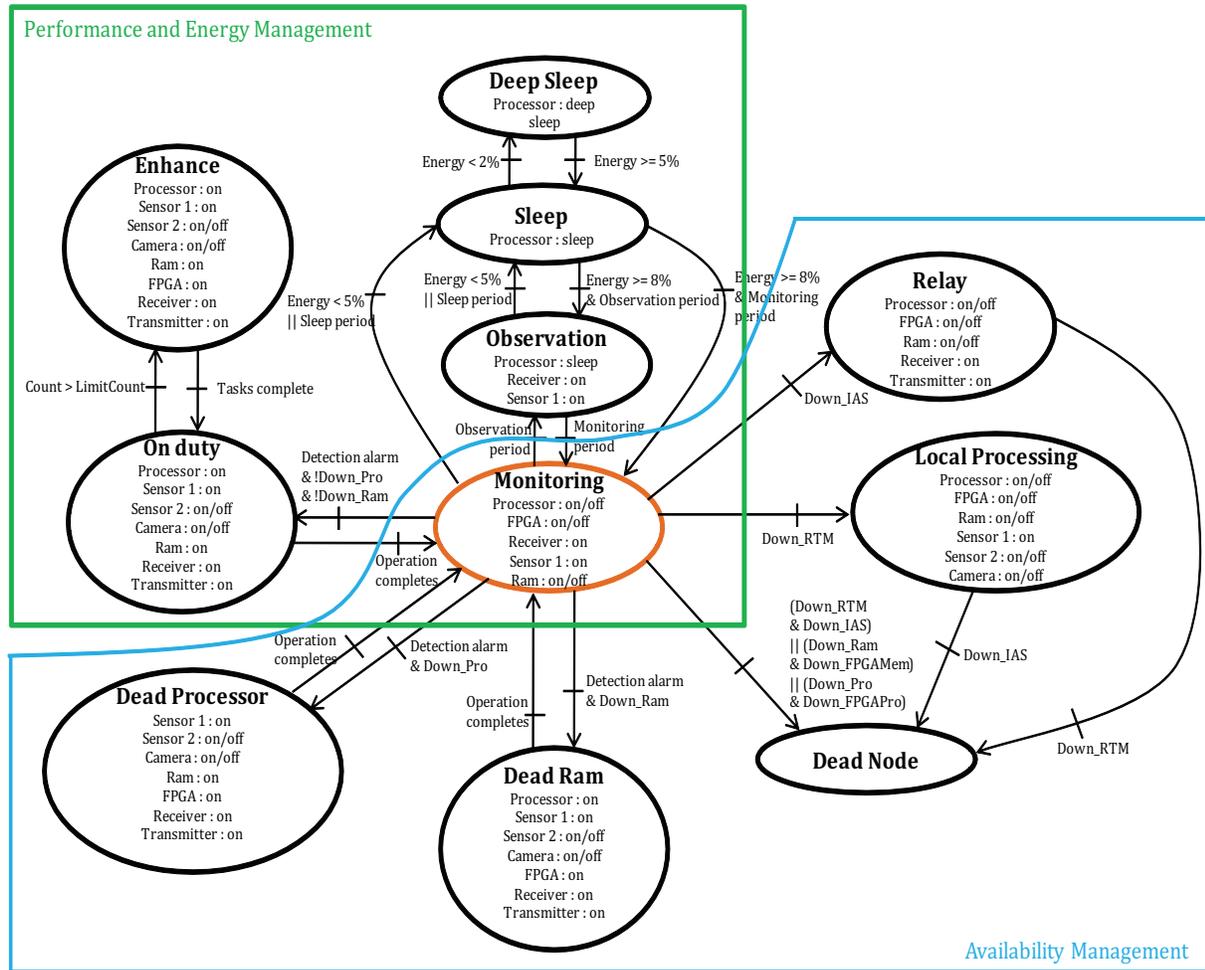


Figure 4.6: FSM model of operating modes for sensor node

by discrete events, the state switches between them. As we know, most of the energy in WSN nodes is consumed by radio transceiver, but all the states (except Deep Sleep and Sleep) in our FSM model have receiver on, because we try to provide a general approach for all applications. For example, in the application of detecting hazardous gaz for the normal area like warehouse, the radio transceiver is only turned on in a time interval based on the generated events. Thus, the sensor node switches mostly between Sleep, Monitoring and On-Duty modes. But with the same application for residential area, the radio receiver is always turn on, even in low power modes, to receive and send rapidly the data to the supervisor in case of detecting dangerous gaz. Therefore, the evacuation can be rapidly executed. In this application, the sensor node switches mostly between Observation, Monitoring and On-Duty modes to save energy.

Besides, the performance of the node system is also considered to reduce the execution time of application. That leads to improve performance of the network. For example, the state

is initially in Monitoring. When alarm detection is generated, state changes to On-Duty, and Sensor2 or Camera can be turned on for image or video processing application. If the execution time of application passes a time deadline, PAM block activates the FPGA that allows parallel processing between processor and FPGA processor. Thus, this leads to enhance execution. After completing all the tasks, the system comes back to Monitoring state.

As previously mentioned, the state of FSM model is initially in Monitoring. When Sensor1 generates a detection alarm and if the main processor is down, state changes to Dead Processor. Consequently, FPGA processor is enabled to replace the main one for processing the data. Some special devices such as Sensor2 or Camera can be turned on to verify the circumstance or take a video of the scene. After completing all tasks, the state comes back to Monitoring state. The procedure is similar when Ram memory is down, the FPGA memory is enabled, and state changes to Dead Ram if an alarm detection arrives. The other problems are considered such as failure of either the IAS or the RTM. The sensor node is considered as a relay point in the first case, or as local processing in second case. The local processing mode is defined because in some applications like detection of hazardous gaz, we keep the sensor node still running even when its radio module is failed. Thus, the technician can recover the recent captured data when he arrives to repair the radio module. If both sensors and radio transceiver, or main and FPGA processor, or RAM and FPGA memory are down, the node state reaches to Dead Node. The next section presents the evaluation of our approach through simulation and implementation results.

4.3 Simulation Experiment

This section shows the simulation results obtained when applying our approach in wireless sensor network. Since no simulation tool that may take both two aspects energy and availability into consideration exists, we must simulate separately these two aspects in two different tools. The detailed description of each tool is provided in the remainder of this section.

4.3.1 Energy-saving

CAPNET Power-Energy estimator (CAPNET-PE) is created and developed by our Lab-STICC laboratory for the CAPNET project [Nicolas Ferry, 2011 [26]] [Nicolas Ferry, 2010 [27]] in cooperation with ERYMA Company. This simulator allows identifying power consumption hot spots and making critical choices during the system design. It also helps to scale the energy storage system as well as the energy harvesters correctly. An example of simulation model of our sensor node is illustrated in Fig.4.7.

As shown in Fig.4.7, the node model is composed of two layers: the logical and the hardware layer. Both are tightly coupled and are synchronized to work together. While the logical level defines the node environment as several external models: scenario events, RF events, meteorology, and weather forecast models. The DPM model describes the energetic strategy developed

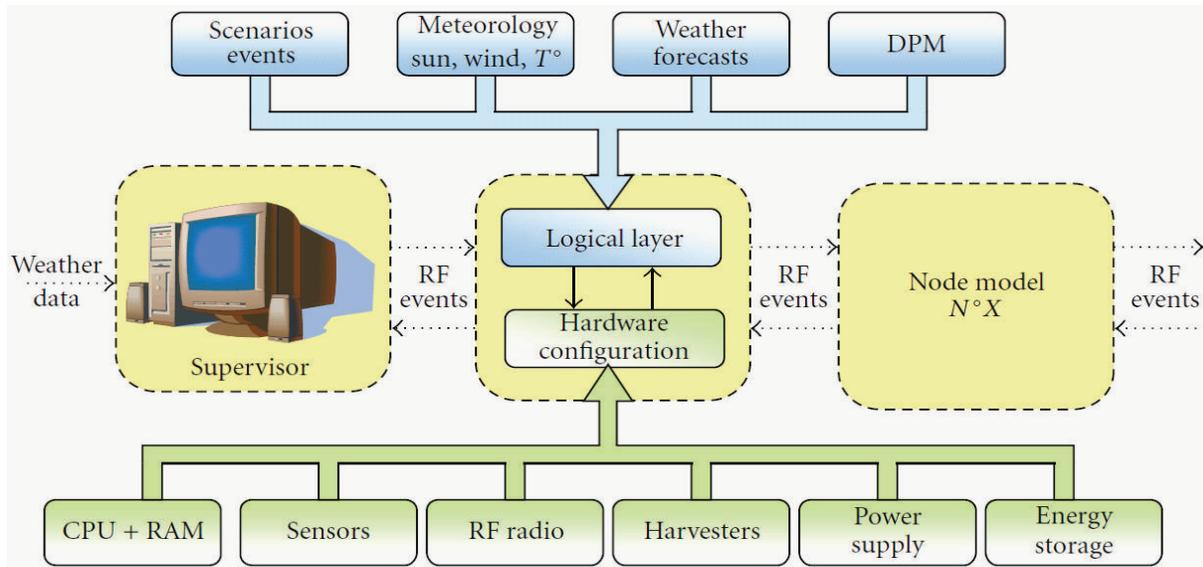


Figure 4.7: Principle of node energy simulation in CAPNET-PE tool

on the node system. The hardware layer is modelled using the FLPA methodology [J. Laurent, 2004 [55]] to develop different node component models with a variable accuracy. Therefore, the simulator is not only to help designers to identify major power consumption hot-spots but also predicts the system autonomy. On the other hand, simulating a DPM permits our node to adjust the strategy and tune consistently DVFS and DPM opportunities in order to increase energy-efficiency. Thus, designers can evaluate the relevance of the developed DPM policy. Then, finally, a realistic estimation of the global node system viability for a configuration is given by this tool.

In the energy simulation, our approach is tested with the application of hazardous gas detection for area such as harbor or warehouse, which is developed by the ERYMA Company [Eryma Co., [75]]. In this application, the operating mode of node system is initially at Monitoring, if no event occurs during a time interval, the system enters in Sleep mode to save energy (as depicted in Fig.4.6). Otherwise, it will be in On-Duty mode if an alarm is detected. The original sensor node consists of a PIC24FJ256GB110 processor, a M48T35AV Ram memory, a Miwi radio transceiver, two Oldham OLCT 80 gas detectors, the power switches (LM3100 and MAX618), and a battery as depicted in Fig.4.8). The power consumption of these components are measured and provided in [Nicolas Ferry, 2011 [27]].

Concerning the choice of FPGA manufacturer, the Actel FPGA IGLOO Nano will be selected to be implemented in our sensor node due to its reliability and ultra-low power consumption [66]. The advantages of Actel FPGA IGLOO Nano are explained in implementation section 4.4.1. Actel manufacturer offers soft-CPU such as ARM Cortex-M1, or core8051s, or coreABC that are available in the Actel library of Intellectual Property (IP) cores. Thanks to

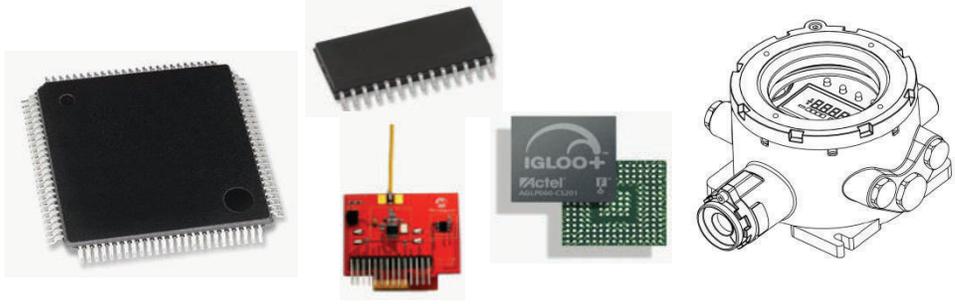


Figure 4.8: The components of wireless sensor node

its good trade-off between performance/energy consumption, the core8051s is selected for performance enhancement. This soft-CPU is only activated when the performance enhancement is needed in case of hazardous gas detection as described in previous section 4.2.3, or to replace the failed processor. Additionally, our PAM block will be also programmed to be implemented in FPGA using small resource because it does not realize any complex computation. Based on the power calculation methodology in IGLOO Nano datasheet [66], the consumed powers of soft-CPU core8051s and our PAM are estimated. The power values of our node components are given in the Table 4.3.

Table 4.3: Power measurement of node component

Component	Supply voltage (V)	Power (mW)
PIC24FJ256GB110	3.3	51
M48T35AV memory	3.3	150
Oldham OLCT 80	20	867
Miwi Radio transmitter	3.3	130
Miwi Radio receiver	3.3	70
Core8051s	1.5	26
PAM block	1.5	0.2

To apply DVFS method without suffering large energy overhead due to its complex computation, two supply voltages of 3V and 3.6V are selected for processor, memory and radio module. These two voltage values correspond to the minimum and maximum functioning voltage noted in the datasheets of these components. Based on the Eq.3.7, their new values of power dissipation are computed as follows:

$$P_t = \frac{V_t^2}{V_{3.3}^2} \cdot \frac{f_t}{f_{3.3}} \cdot P_{3.3} = \frac{V_t^2}{V_{3.3}^2} \cdot \frac{1}{e} \cdot P_{3.3} \quad (4.2)$$

Where P_t and V_t are respectively the actual power consumption and supply voltage, and $e = \frac{T_t}{T_{3.3}}$ is extension factor between the new and nominal time execution ($e_{3/3.3} = 1.17$ and $e_{3.6/3.3} = 0.87$ in our study). To simulate the energy dissipation using CAPNET-PE tool, we

must create a project. After completing the project creation steps, we configure all available components of sensor node in this project. Finally, the sensor node model implemented in CAPNET-PE interface is similar to Fig.4.9.

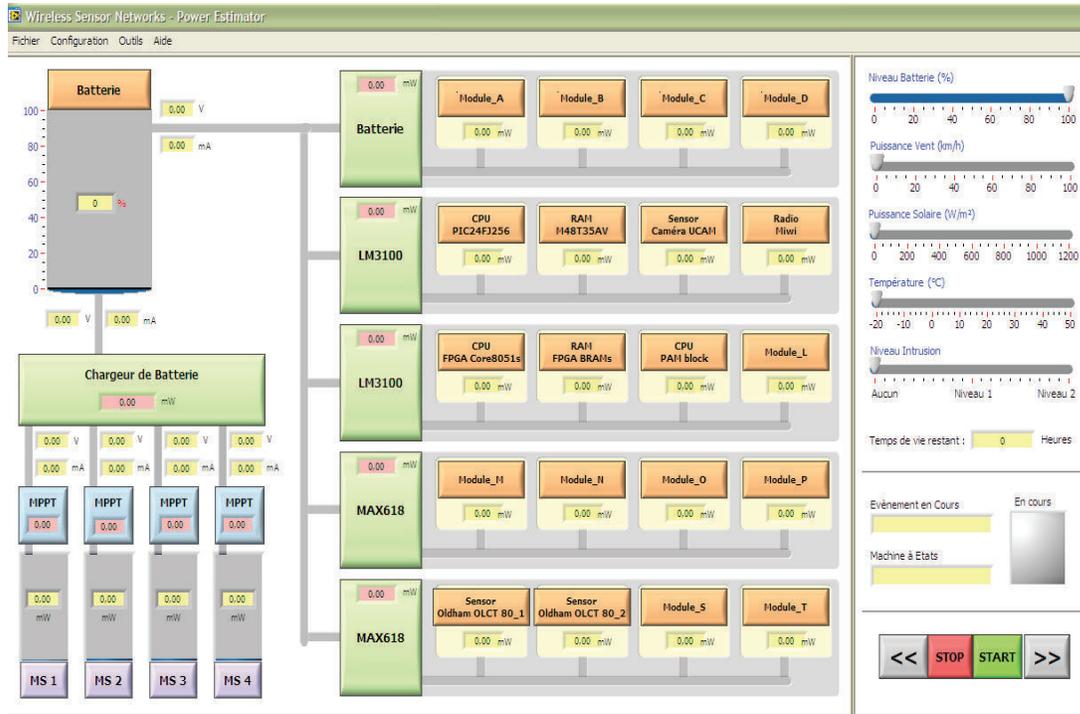


Figure 4.9: Block diagram of sensor node in CAPNET-PE tool

A scenario of energy simulation is defined in CAPNET-PE as depicted in Fig.4.10 for this application. This scenario is composed of two events which control the entire sequence. In the first one, the gas sensor is turned on to measure but does not detect any hazardous gas. And the second one indicates that there is a detection of hazardous gaz. Each event signal has its start time, the duration, the periodicity for periodic events, and the numbers of repetitions.

Based on this event scenario, the operating functionality of each node component is programmed in DPM policy tab of CAPNET-PE. These events are consumed by the DPM policy which verifies the rules at each discrete time and consequently performs the corresponding actions. For example, the code prototype of DPM policy when applying DPM and DVFS method is given below.

```
// Initialize the number of node devices
NbDeviceOut <- NbDeviceIn;

// Initialize the operating mode of node devices
for (int i=0; i<=NbDeviceIn; i++){
  Mode_Device_Out[i] <- Mode_Device_In[i];
}
```

Event Name	Priority	Nature	Type	Start	Period	Duration	Resolution	Repeat
Detection of Hazardous Gaz	1	Measure	Periodic	10	28800	53	seconds	21
Measure Gaz	2	Measure	Periodic	10	900	53	seconds	672

60 in case of DPM+DVFS

Figure 4.10: Event scenario of energy simulation

```
// Select the operation mode based on the event type
switch (Mode_In){
  // Initialization Mode
  case 0:
    State_After_Out <- 0; // State after timer counter
    Timer_Out <- 0; // Timer sets to 0
    Mode_Device_Out[Processor] <- Sleep;
    Mode_Device_Out[Ram] <- Off;
    Mode_Device_Out[Camera] <- Off;
    Mode_Device_Out[Radio] <- Off;
    Mode_Device_Out[FPGAPro] <- Off;
    Mode_Device_Out[PAM] <- Off;
    Mode_Device_Out[Sensor1] <- Off;
    Mode_Device_Out[Sensor2] <- Off;

    Mode_Out <- 1; // operating mode switches then to Sleep mode
    break;

  // Sleep Mode
  case 1:
    switch (Event_In){
      // No Event
      case 0:
        Mode_Device_Out[Processor] <- Sleep;
        Mode_Device_Out[Ram] <- Off;
        Mode_Device_Out[Camera] <- Off;
        Mode_Device_Out[Radio] <- Off;
        Mode_Device_Out[FPGAPro] <- Off;
        Mode_Device_Out[PAM] <- Off;
        Mode_Device_Out[Sensor1] <- Off;
        Mode_Device_Out[Sensor2] <- Off;

        Mode_out <- 1;
        break;

      // Generated event of no detection
      case 1:
        Mode_Device_Out[Processor] <- Pro_Low_Power;
        Mode_Device_Out[Ram] <- Ram_Low_Power;
        Mode_Device_Out[Camera] <- Off;
        Mode_Device_Out[Radio] <- Radio_Low_Power;
        Mode_Device_Out[FPGAPro] <- Off;
        Mode_Device_Out[PAM] <- On;
        Mode_Device_Out[Sensor1] <- Off;
        Mode_Device_Out[Sensor2] <- Off;
    }
  }
}
```

```

    Mode_Out                <- 2; //Turn on the first sensor
    break;

    // Generated event of gas detection
    case 2:
        Mode_Device_Out[Processor] <- Pro_High_Power;
        Mode_Device_Out[Ram] <- Ram_High_Power;
        Mode_Device_Out[Camera] <- Off;
        Mode_Device_Out[Radio] <- Radio_High_Power;
        Mode_Device_Out[FPGAPro] <- Off;
        Mode_Device_Out[PAM] <- On;
        Mode_Device_Out[Sensor1] <- Off;
        Mode_Device_Out[Sensor2] <- Off;

        Mode_Out                <- 2; //Turn on the first sensor
        break;

    default:
        break;
}

// Turn on the sensor 1
case 2:
    Mode_Device_Out[Sensor1] <- On;
    switch(Event_In){
        // Generated event of no detection
        case 1:
            Timer_Out          <- Measure_With_No_Detection_Time;
            State_After_Out    <- 5;
            break;

        // Generated event of gas detection
        case 2:
            Timer_Out          <- Measure_With_Detection_Time;
            State_After_Out    <- 3;
            break;

        default:
            break;
    }
    Time_Init                <- Time; // Save the actual time of system
    Mode_Out                  <- 6;
    break;

// Turn on camera and second sensor to verify the alarm of gas detection
case 3:
    Mode_Device_Out[Sensor1] <- Off; // Turn off the first sensor
    Mode_Device_Out[Cam] <- On;
    Mode_Device_Out[FPGAPro] <- On; // Turn on FPGAPro to allow parallel execution
    Mode_Device_Out[Sensor2] <- On;
    Timer_Out                <- Time_to_turn_off_camera;
    Time_Init                 <- Time; // Save the actual time of system
    State_After_Out          <- 4; // Turn off camera+FPGAPro
    Mode_Out                  <- 5;
    break;

// Turn off the sensor and FPGAPro
case 4:
    Mode_Device_Out[Cam] <- Off;
    Mode_Device_Out[FPGAPro] <- Off;
    Timer_Out            <- Time_to_turn_off_second_sensor;
    Time_Init            <- Time; // Save the actual time of system
    State_After_Out      <- 5; // Turn off camera+FPGAPro

```

```

Mode_Out          <- 6;
break;

// Turn off all component and back to Sleep mode
case 5:
Mode_Device_Out[Processor] <- Sleep;
Mode_Device_Out[Ram]      <- Off;
Mode_Device_Out[Camera]   <- Off;
Mode_Device_Out[Radio]    <- Off;
Mode_Device_Out[FPGAPro]  <- Off;
Mode_Device_Out[PAM]      <- Off;
Mode_Device_Out[Sensor1]  <- Off;
Mode_Device_Out[Sensor2]  <- Off;
Timer_Out              <- 0;
Mode_Out               <- 1;
break;

// Time Counter
case 6:
State_After_Out        <- State_After_In; // Save the output state
if((Time - Time_Init) <= Timer_Out){
    Mode_Out <- 5;
} else {
    Mode_Out <- State_After_In;
}
break;

default:
break;
}

```

Listing 4.1: Code Prototype of DPM policy

To show the advantage of our approach, three cases are realized by using CAPNET-PE tool as follows:

1. Neither PAM block combined with FPGA nor two methods of DPM and DVFS are used.
2. Only use of PAM block combined with FPGA, and DPM method. The FPGA Core8051s is used for image processing in case of detection of hazardous gas.
3. Both use of PAM block combined with FPGA, DPM and DVFS methods. The supply voltage and the operating frequency are dynamically regulated for each component based on the type of generated event. To mitigate the large overheads in computation by using DVFS, the supply voltage is set maximum to components such as processor, Ram memory, and radio module in case of hazardous gas detection, otherwise they are supplied with minimum voltage when any detection is found.

We evaluate the impact of our approach in node energy dissipation in function of the rate of generated discrete events per day. Three same cases are realized with duration of one day using CAPNET-PE tool. The energy consumptions of processor, memory and radio module are respectively depicted in Fig.4.11. This figure indicates that when DPM method is used, a large significant energy-saving up to 75% for processor, and 32% for memory when the number of event is high (240 events per day), because the consumed powers when these components are in

Idle mode are much higher than the powers when they are in *Deep-Sleep* mode. The simulation results of third case still show an improvement in energy reduction compared to the second case, when energy gains are up to 80% for processor, 53% for memory, and 31% for radio module. The Fig.4.11 shows the percentage of energy reduction using DPM method or both DPM and DVFS methods in wireless sensor node, in which energy gains are up to 69% and 72% when number of event is low (24 events per days). But these gains are degraded when the number of event is high, only 25% and 35% of energy reduction, because the gas sensor of node, which is the most critical element in term of energy consumption (as listed in Tab.4.3), is enabled more times to sense data than the cases of lower number of event. This energy dissipation is unavoidable.

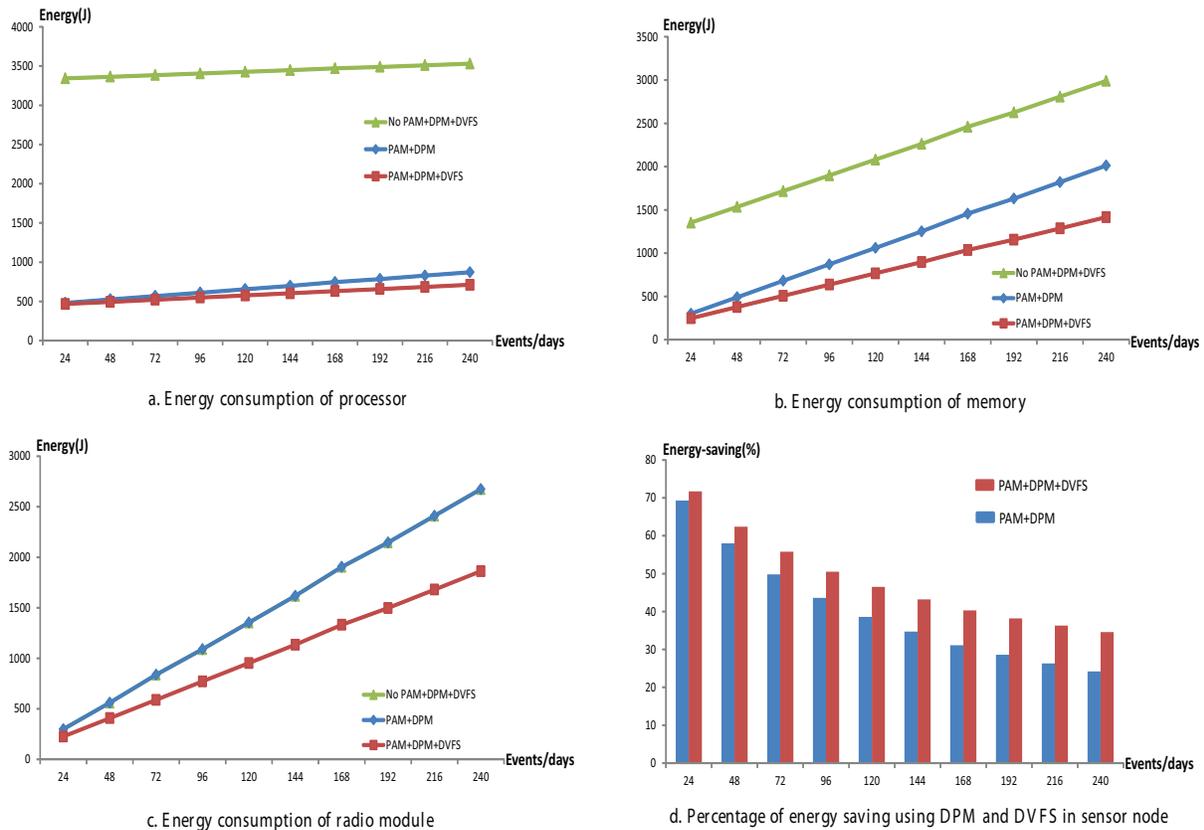


Figure 4.11: Energy consumption versus event rate

However, we also show the interest of implementing DPM and DVFS methods in sensor node through the energy gains in node components such as processor, memory and radio module. Additionally, the consumption of both PAM block and FPGA is very small as depicted in Fig.4.12, which makes them suitable to implement into sensor node. The next sub-section shows our evaluation in term of node availability when applying our solution.

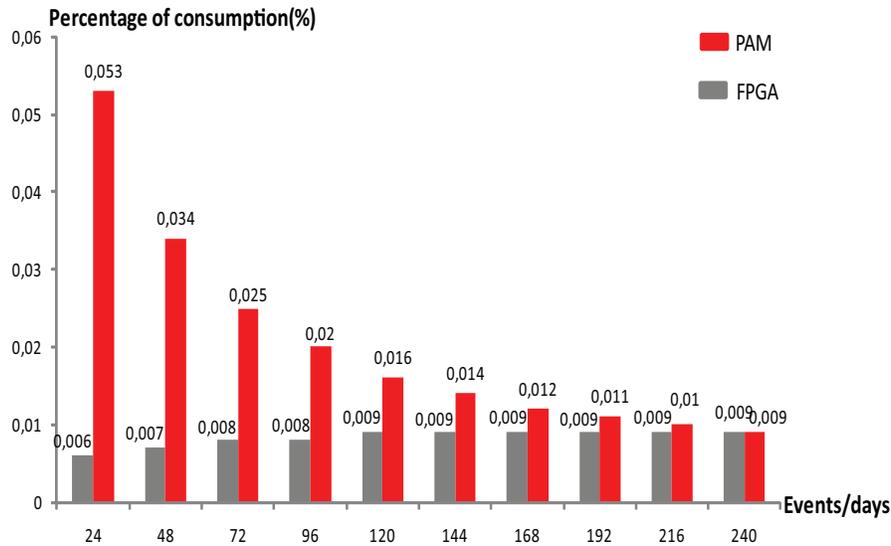


Figure 4.12: Consumption percentage of PAM and FPGA in wireless node

4.3.2 Availability improvement

The Stochastic Petri Net Package (SPNP) tool presented in [J. Muppala, 1989 [28]] as described in Fig.4.13, is used for computing the availability of our sensor node. This tool is created and developed by Professor Kishor S. Trivedi and al. at Duke University. It offers tool for Petri Net modeling and simulation of Discrete Event System (DES). This tool is used in many applications to compute the failure probability, the reliability, or the availability of DES system.

We now introduce Mean Time To Failure (MTTF) for each node component.

Definition 4.3.1. *Mean Time To Failure (MTTF) is defined for non-repairable systems to indicate the average functioning time from instance 0 to the first appearance of failure.*

These MTTFs are shown in Tab.4.4. In this section, we do not simulate the software bug problem, we focus on the failure of each component that is more serious. The time to occurrence of failure in the Sensors, Ram memory, Processor and Radio module is assumed to be random variable with exponentially distributed rates λ_s , λ_{ram} , λ_p and λ_{radio} , which are constant.

Since the failed component in our node cannot be repaired, its availability is computed as same as its reliability computation, $A(t) = R(t)$, where $R(t)$ is computed as follows:

$$\lambda(t) = \frac{1}{R(t)} \cdot \frac{dF(t)}{dt} = \frac{1}{R(t)} \cdot \frac{-dR(t)}{dt} \quad (4.3)$$

In which, $F(t)$ is the failure probability of component at the time t , thus:

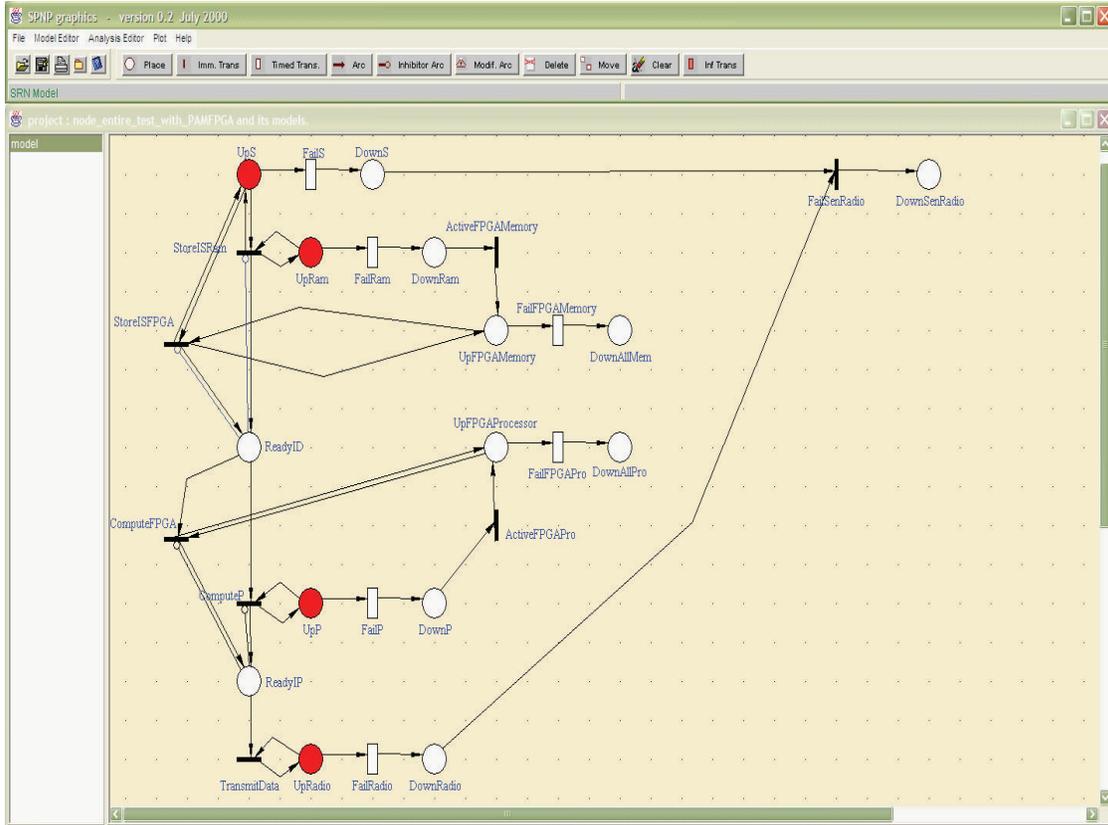


Figure 4.13: User interface of SPNP tool

$$\lambda(t)dt = -\frac{dR(t)}{R(t)} \tag{4.4}$$

Inferring that:

$$\int_0^t \lambda(x)dx = -\int_0^t \frac{dR(x)}{R(x)} = -\ln R(t) \tag{4.5}$$

As $R(0)=1$ and $\ln(1)=0$, thus $R(t) = e^{-\int_0^t \lambda(x) dx} = e^{-\lambda.t} = A(t)$, because the failure rate is constant. Based on the MTTF values of node components, their failure rate values are computed as following:

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda.t}dt = \frac{1}{\lambda} \tag{4.6}$$

Thus, $\lambda = \frac{1}{MTTF}$, the failure rates λ_s , λ_{ram} , λ_p and λ_{radio} are given in Tab.4.4.

The failure probability of each component is computed as follows:

Table 4.4: Failure Rate and MTTF for each component

Component	Mean Time To Failure	Failure rate (λ)
Sensor	MTTF of a sensor is 3.4 years	1/30000 fail/hour
Processor	MTTF of processor is 30 years	1/262800 fail/hour
RAM	MTTF of RAM memory is 9.5 years	1/4320 fail/hour
RTM	MTTF of radio transceiver is 11.4 years	1/100000 fail/hour
FPGA memory	MTTF of FPGA memory is 9.1 years	1/80000 fail/hour
FPGA processor	MTTF of FPGA processor is 15 years	1/131400 fail/hour

$$F_{sensors} = (1 - A_{sensor1}) * (1 - A_{sensor2}) \quad (4.7)$$

$$F_{processor} = 1 - A_{processor} \quad (4.8)$$

$$F_{ram} = 1 - A_{ram} \quad (4.9)$$

$$F_{radio} = 1 - A_{radio} \quad (4.10)$$

In the equation 4.7, since our sensor node is equipped two sensor, while the first one is usually enable to capture environment data, the second one is only activated by users to verify the data accuracy. Moreover, in case of failure of first sensor, the second one can be enable to replace it. Since the failure of two sensor is independent, hence the failure probability of node sensors is the multiplication of failure probability of each sensor.

The Time Stochastic Petri Net (TSPN) model of our node functionality is depicted in the Fig.4.14, where the failure localization model of each component is included in the transitions $T:FailS1$, $T:FailS2$, $T:FailP$, $T:FailFPGAP$, $T:FailRam$, $T:FailFPGAMem$, and $T:FailRadio$. The failure rates of node components are respectively depicted by exponentially distributed firing rates λ_s , λ_p , λ_{fpgap} , λ_{ram} , $\lambda_{fpgamem}$, λ_{radio} (see Table 4.4). As seen in this figure, the immediate transitions are modelled as black bars, when others are modelled as timed transition (white rectangles).

When a data captured by sensors is stored in buffer, a token is present in $P:ReadyDS$. The inhibitor arc from one place to a transition means that the firing of this transition is prohibited when a token appears in this place. An inhibitor arc with multiplicity of 10 from $P:ReadyDS$ to $T:CapD1$ and $T:CapD2$ prevents the number of tokens in this place from being greater than 10, because capturing data buffer size is fixed to 10. Then, the data is processed by processor, if there is a software bug with rate λ_{bug} during execution, the processor is rebooted. When processing is completed, a token is present in the $P:ReadyDP$ meaning that data are ready for storing in memory before transmitting $P:DataForTransmit$.

As depicted in Fig.4.14, appearance of a token in the places $P:DownS1$, $P:DownP$, $P:DownRam$ and $P:DownRadio$ indicates the failure of node components as sensor, processor, Ram memory

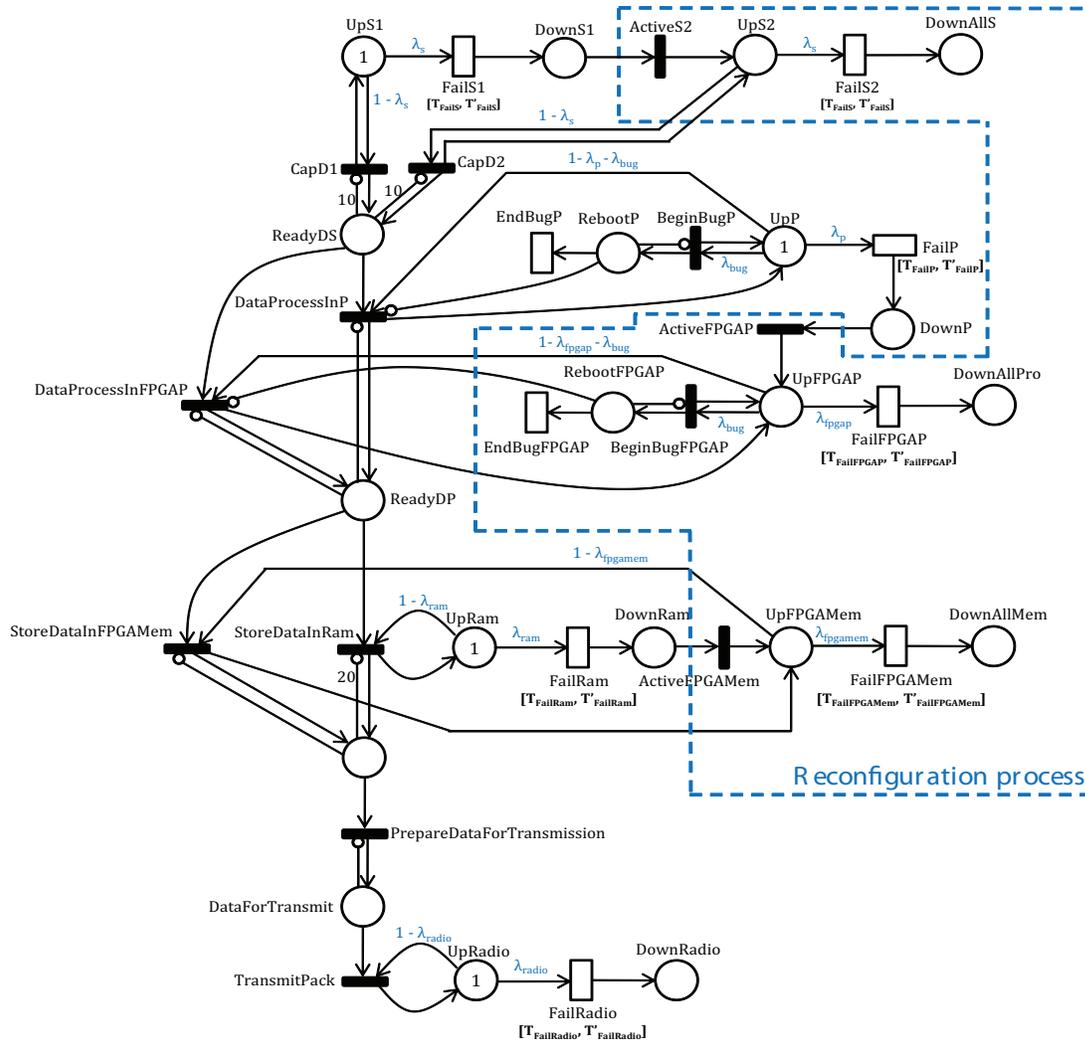


Figure 4.14: TSPN model of sensor node functionality

and radio module. In the figure of node functionality modeling, our reconfiguration approach in case of hardware failure detection is surrounded by dashed blue line. For example, if the first sensor device is down (a token appears in the place $P:DownS1$), the second one is activated to undertake data sensing by distributing a token in $P:UpS2$. Or in case of failure of Ram memory or processor, FPGA memory or FPGA processor is activated for replacing these failed components. The failure rates of FPGA memory and processor are respectively $\lambda_{fpgamem}$, λ_{fpgap} (see Tab.4.4). For radio module failure, the operating mode of node system is automatically switched to Relay mode because we assume that our sensor node is equipped with only one radio module. By using material reconfiguration, the node system is only down when both $P:DownAllS$ and $P:DownRadio$, or $P:DownAllMem$, or $P:DownAllPro$ contain a token. It means that both sensing and radio components, or both Ram memory and FPGA memory, or both main Processor and FPGA Processor are down.

The Fig.4.15 illustrates the failure probabilities of components over ten years, in which the sensors are the most critical components due to their low reliability, because their circuitry is very complex. The curves obtained in the Fig.4.15 are consistent to the results obtained by theoretical computation of failure probability 4.7, 4.8, 4.9 and 4.10, which allows validating our TSPN modeling of node operation.

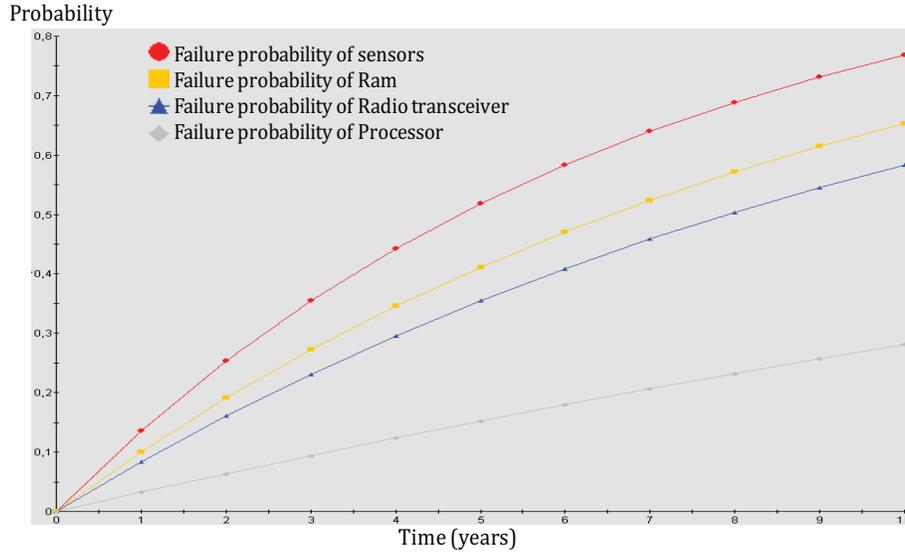


Figure 4.15: Failure probability of node components

Thanks to SPNP tool, we can define the appropriate reward rates to compute the output measures of interest. The advantage is that we need only specify the reward rates associated with certain conditions of the system, instead of explicitly identifying all its states. In our case, the availability of node system is the output measure of interest. Node system is still available if there is not any token in both $P:DownAllS$ and $P:DownRadio$, or $P:DownAllMem$, or $P:DownAllPro$ (see Figure 4.14). To compute the availability of node system by using SPNP tool, we need only specify a reward rate associated with the condition of node availability as follows:

$$r_{availability} = \begin{cases} 0 & \text{if } (\#(DownAllS)=2 \text{ and } \#(DownRadio)=1) \text{ or} \\ & \#(DownAllMem)=1 \text{ or} \\ & \#(DownAllPro)=1. \\ 1 & \text{otherwise.} \end{cases} \quad (4.11)$$

Where: $\#(p)$ represents the number of tokens in place p .

The availability of our node at time t is computed as the expected instantaneous reward rate $E[X(t)]$ at time t , where $X(t)$ is a random variable corresponding to the instantaneous reward rate of node availability. The expression of $E[X(t)]$ is described as follows:

$$E[X(t)] = \sum_{k \in T} r_k \cdot \pi_k(t) \quad (4.12)$$

$\pi_k(t)$ is the probability of being in marking k at the time t , and T is the set of markings. The computation of $\pi_k(t)$ is described in detail in [G. Ciardo, 1993 [24]]. Based on this computation, our node availability is depicted in the Fig.4.16. An availability threshold of 90% is selected for node to guarantee their correct functionality. The Fig.4.16 shows the node availability with and without our reconfiguration approach. As seen in this figure, when the node availability is not less than 90%, its operating time increases significantly from 273 *days* (without reconfiguration) to 3.4 *years* (with reconfiguration), which is up to 453%. This significant improvement of node availability shows the interest of material reconfiguration approach for wireless sensor node.

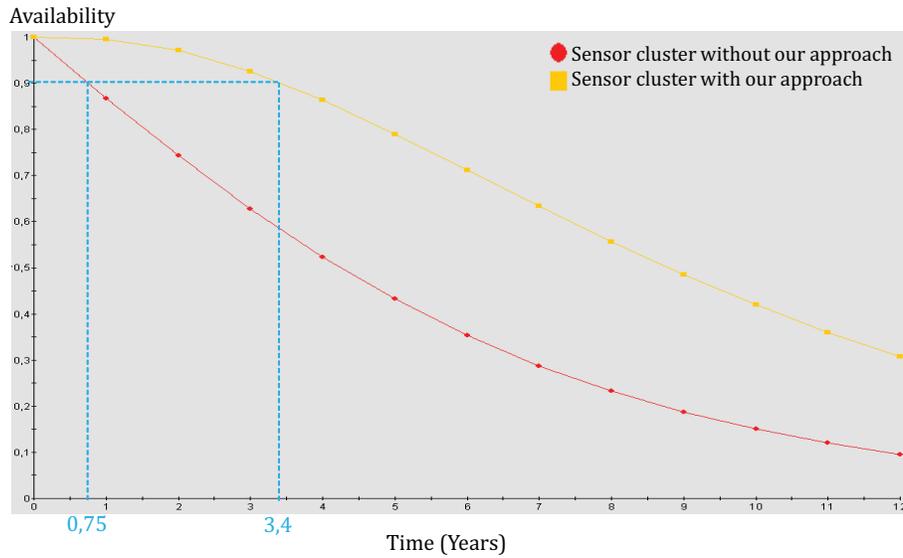


Figure 4.16: Comparison of node availability with and without our approach

We also analyze the impact of node availability improvement in cluster level. We assume that our cluster consists of five sensor nodes including a Gateway and four source nodes as depicted in Fig.4.17.a. As explained previously in section 2.1, the Gateway undertakes the data collection from all the source nodes of its cluster, then sends these data to Sink. Therefore, the cluster is assumed to be unavailable if its Gateway is down or both four source nodes are down. The Fig.4.17.b depicts cluster availability via reliability diagram.

Through the diagram above, the cluster availability is computed as following:

$$A_{cluster} = A_{Gateway} \times (1 - (1 - A_{SourceNode})^4) \quad (4.13)$$

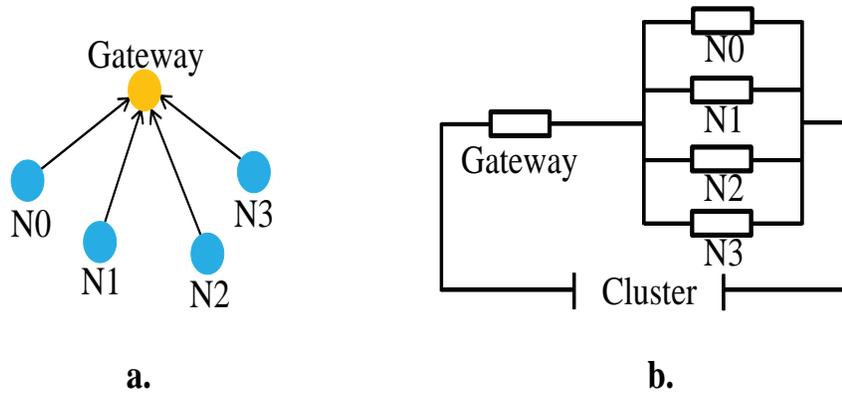


Figure 4.17: Structure and reliability diagram of cluster

Where $A_{Gateway} = A_{SourceNode} = A_{Node}$ that is given in the Fig.4.16. Based on the simulation result of node availability, the cluster availability with and without reconfiguration approach is computed by following to Eq.4.13, which is given in the Fig.4.18.

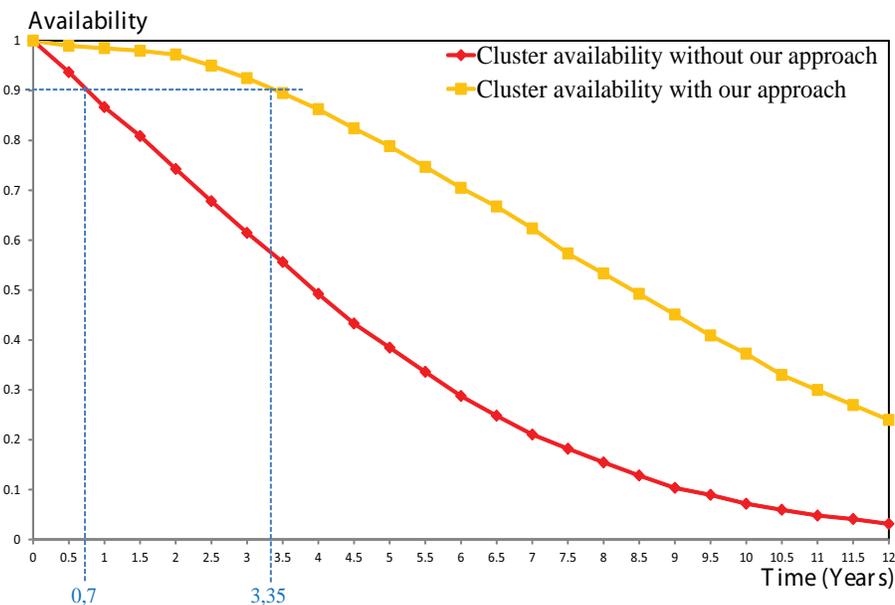


Figure 4.18: Comparison of cluster with and without our approach

As seen in this figure, when the cluster availability is not less than 90%, its operating time increases significantly from 238 days (without reconfiguration) to 3.35 years (with reconfiguration), which is up to 513%. We conclude that our reconfiguration approach improves highly the node availability that leads to increase highly cluster availability.

4.4 Implementation Experiment

This section presents several implementations in real material to demonstrate the feasibility of our approach. In this section, we firstly introduces the implementation of Power and Availability Manager (PAM) in microcontroller and then in FPGA to compare the advantage and disadvantage of each solution. Next, a test-bench for the adjustable regulator is realized to describe how the supply voltage is regulated for node components (processor, memory, radio module). The adjustable regulator realization is a crucial element to apply the DVFS technique in sensor node. This test-bench shows how the voltage level values switch between them to power the node components. Its power consumption and switching time are also measured. Finally, a test-bench for hardware failure detection and localization is realized.

4.4.1 PAM material implementation

PAM implementation in 16-bit microcontroller

A simple application is used to test the implementation of our PAM in real material. The principle of PAM functionality is depicted by a FSM model in Fig.4.19, in which the PAM has three main operating states: low-power state (idle, sleep or deep sleep), active state and execution state. As illustrated in this figure, the PAM spends most of its time in low-power state, it is only waken-up by any interruption arrival and then executes a piece of code, called the Interrupt Service Routine (ISR), directly associated with this interruption, to provide the required response program. If multiple sources of interrupts are used by an application like WSN application in our case, assigning different priority levels to each source is crucial. The priority decides who gets served first if two or more interrupt events happen simultaneously. For example, Microchip microcontroller has up to seven levels of interrupt priority. The more critical the interrupt event is, the more priority it has to be executed. If two interrupt events occur at the same time, the highest priority event will be served first. When one of the interrupts is being served, the others will have to wait for their turn to be served. However, in some cases the application requires not only multiple interrupts but the ability to nest the interrupt calls. When a lower-priority interrupt is being served and the ISR is being executed, an arrived higher-priority interrupt might require immediate attention. In this case, the ISR execution of lower-priority interrupt will be immediately preempted in order to execute the ISR of the higher-priority interrupt. When the ISR execution of higher-priority interrupt is completed, the program is back to preempted ISR of lower-priority interrupt.

In this test of PAM implementation, only one external interruption source (change notification interrupt at a port) is used to test the functionality of PAM. When this interrupt event occurs, the PAM will enter in active state from low-power state, and then execute the corresponding ISR code (Led blink in our test).

At the first attempt, the PAM is implemented in a microcontroller. In the world of electronics, the term microcontroller is very widely used. Almost every single electronic device always

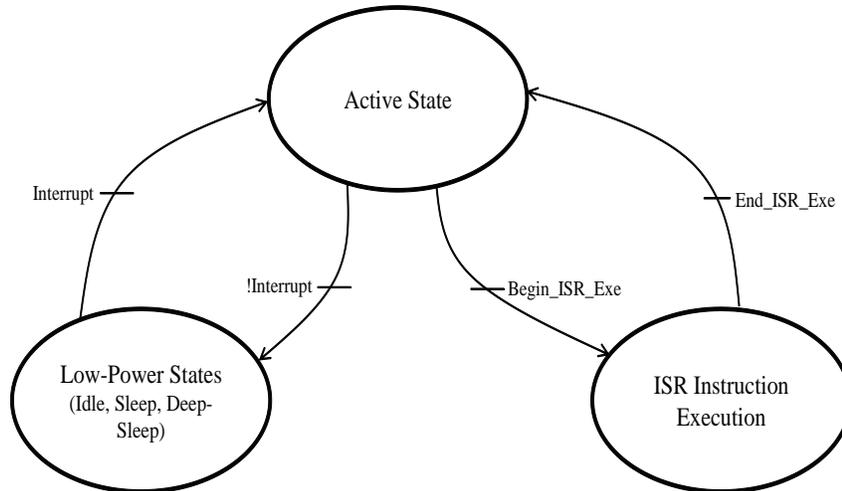
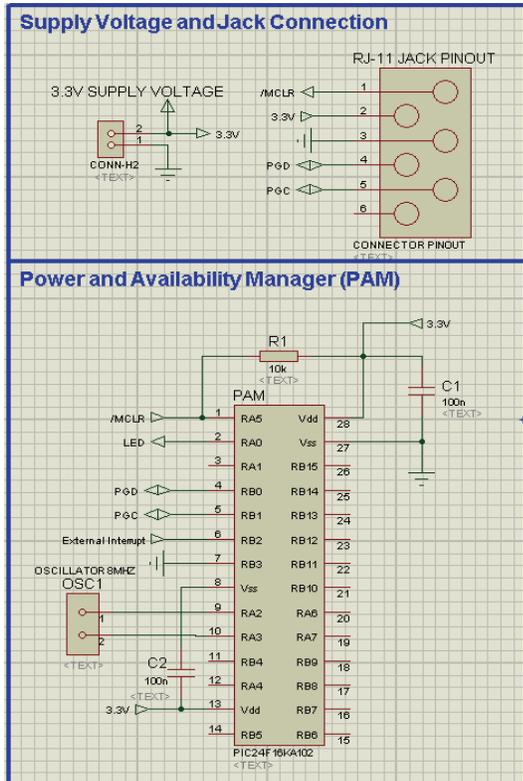


Figure 4.19: Principle of PAM functionality

has an embedded microcontroller inside to facilitate the communication with the computer. The internal structure of a microcontroller looks similar to a simple computer placed in a single chip with all of the necessary components like memory and timers embedded inside. It is programmed to do some simple tasks for other hardware. In this test, a Microchip PIC24F16KA102 16-bit flash microcontroller with nanoWatt XPL technology is selected to implement the PAM. Microchip is an American manufacturer of microcontroller, memory and analog semiconductors, with low-risk product development, low-power consumption and lower total system cost. The C program code that describes the PAM functionality is written in MPLAB development environment of Microchip. This development tool is free for downloading and using for academy.

To design the connection between the blocks in the test-bench, the Proteus tool is used. This tool allows not only designing the block connection using Proteus ISIS but also mapping and routing component blocks in a pre-defined board using Proteus ARES, in order to provide the schema of Printed Circuit Board (PCB). As illustrated in Fig.4.20, the PAM microcontroller is placed in complementary testing board connecting to socket connector of Explorer16 development board [71]. The application program is then transferred from computer to PAM microcontroller via RJ-11 jack connector to run application.

The PAM is powered by 3.3V supply voltage, and its frequency is configured to run at 4MHz by using an 8MHz quartz oscillator. The measure of power consumption is shown in Tab.4.5. The PAM microcontroller consumes 16.2mW at active state, 17.2mW at ISR execution state, and only 0.17mW at sleep state. Since our PAM spends most of its time in sleep state, hence it consumes very small amount of sensor node energy. It should be noted that the power consumption of PAM in sleep mode differs in function of the number of active peripherals, but this power gap is very small. The wake-up time overhead of the PAM microcontroller when



Block diagram of PAM in Proteus ISIS tool

Testbench of PAM implementation in microcontroller

Figure 4.20: PAM implementation in Microchip microcontroller

switching from sleep state to active state is also measured. Based on technical document, this 16-bit microcontroller requires $3T_{clk} = 0,75\mu s$ to exit the sleep state. That allows the PAM to react promptly against any issue occurring inside sensor node.

Table 4.5: Measured power consumption of PAM microcontroller

Component	State	Supply Voltage(V)	Current(mA)	Power(mW)
PAM (PIC16)	Sleep	3.3	0.05	0.17
	Active	3.3	4.9	16.2
	Active (led)	3.3	5.4	17.9

PAM implementation in FPGA IGLOO

A Field Programmable Gate Array or FPGA is an integrated circuit designed to be configured by a customer or a designer after manufacturing - hence "field-programmable". Every single FPGA may contain millions of logic gates and large resources of RAM block to implement complex digital computations. Besides, the FPGAs contain programmable logic components called "logic blocks" that are wired together. Logic blocks can be configured to perform complex

combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. This very basic nature of FPGAs allows it to be more flexible than most microcontrollers. The programmers can reprogram the FPGA device by rewriting all the logic gates to do any tasks that are fitted to the number of gates available in this FPGA device. Vice versa, microcontrollers already have their own circuitry and instruction set that the programmers must follow in order to write code for that microcontroller, which restricts it to certain tasks. It seems that FPGAs are going to rule in the future because of their flexibility, increasingly better power efficiency and decreasing prices. Therefore, after the PAM implementation test in microcontroller, our technological trend turns to implement the PAM in FPGA.

There are several FPGA vendors like Altera, Xilinx, Atmel, Lattice, Actel in commercial market. Our Lab-STICC laboratory has experienced in FPGA Xilinx and Altera through several projects. One of which is OPEN-People project that is funded by the ANR (Agence Nationale de la Recherche). This project aims to provide a hardware/software platform to facilitate estimation and optimization of power and energy consumption for any FPGA-based applications. Three FPGAs are used in this project to realize power consumption measurement such as Virtex-5 and Spartan-6 of Xilinx vendor, and Cyclone-III of Altera vendor. These two FPGA vendors are two largest enterprises that are specialized in the development and commercialization of programmable logic devices. Through experiment results, these three FPGAs offer a good performance but they come at the expense of highly dissipated energy. Since our sensor node has a limit energy budget, these FPGA types are not suitable ones in our study. When making a searching in FPGA commercial market, the IGLOO family of flash FPGA of Actel vendor gets our attention.

The FPGA IGLOO [66] based on 130-nm flash process, offers the lowest power FPGA, small foot print package, reprogrammability. Alongside with the trend to improve better energy-efficiency for FPGA, new Flash*Freeze technology available in IGLOO devices allow FPGA to enter and exit ultra-low power Flash*Freeze mode, which consumes nanoPower while retaining SRAM and register data. Particularly, the IGLOO devices do not need additional components to turn off I/Os or clock while retaining design information. The power management through I/O and clock management in Flash*Freeze mode is simplified to have a rapid recovery to operation mode. The Low Power Active capability (static idle) allows for ultra-low power consumption while the IGLOO device is completely functional in the system. Additionally, the Actel IGLOO devices offer 1 kbit of on-chip, reprogrammable, nonvolatile FlashROM memory, which allows having not only quick loading of device configuration at power-up, but also the advantage of being a secure, low power and single-chip solution. The IGLOO clock can be easily regulated from a clock generator (20MHz quartz oscillator) based on an integrated phase-locked loop (PLL). After a careful review in FPGA IGLOO family, the IGLOO nano is selected for our study. IGLOO nano devices have up to 250 k system gates, supported with up to 36 kbits of true dual-port SRAM and up to 71 user I/Os. These FPGAs increase the breadth of the IGLOO product line by adding new features such as smaller footprint packages, power consumption

measured in nanoPower, Schmitt trigger, and bus hold (hold previous I/O state in Flash*Freeze mode) functionality make these devices ideal for WSN applications that require high levels of flexibility and low cost. Tab.4.6 shows the features through the technical documents of Virtex-5 and Spartan-6 of Xilinx vendor [J.T. Rasolofonirina, 2012 [57]], Cyclone-III of Altera vendor [Lucile Senn, 2011 [58]], and IGLOO nano device of Actel vendor.

Table 4.6: Feature description of FPGAs

Device Feature	Xilinx		Altera	Actel
	Virtex-5	Spartan-6	Cyclone-III	IGLOO Nano
Logic blocks	4800 - 30720	600 - 23038	5136 - 198464	260 - 6144
D Flip-Flops	19200 - 122880	4800 - 184304	5136 - 198464	260 - 6144
Distributed-RAM (Kb)	320 - 2280	75 - 1355	NA	0 - 36
DSP Slices	32 - 384 (25x18)	8 - 180 (18x18)	23 - 396 (18x18)	NA
BRAM Blocks (Kb)	1152 - 16416	216 - 4824	414 - 8019	0 - 36
Flash-ROM (bits)	NA ¹	NA	NA	1024
Max user IOs	400 - 960	132 - 540	182 - 429	34 - 77
PLL (clock management)	1 - 6	1 - 6	2 - 4	0 - 1
Hardcore	0 - 2	NA	NA	NA
Length x Width (mm)	19x19 - 42.5x42.5	8x8 - 21x21	8x8 - 34.6x34.6	3x3 - 14x14
Power static (mW)	443(XC5VLX50T)	31(XC6SLX45T)	150(EP3CLS200F7)	0.009(AGLN250V2)
Price/unit (dollar)	218	60	32	20

Based on feature description specified in Tab.4.6, the advantages and disadvantages of each FPGA are shown in Tab.4.7.

Table 4.7: Advantages and disadvantages of FPGAs

Device Feature	Xilinx		Altera	Actel
	Virtex-5	Spartan-6	Cyclone-III	IGLOO Nano
Logic resources	++++	+++	++++	+
Memory resources	++++	+++	++	+
DSP slices	+++	++	+++	NA
BRAM Blocks (Kb)	++++	++	+++	-
Flash-ROM (bits)	NA	NA	NA	++
Max user IOs	++++	++	++	+
PLL resources	+++	+++	++	+
Hardcore	++	NA	NA	NA
Size dimension	+	++	+	+++
Power dissipation	----	+	--	++++
Price/unit	--	+	++	+++

Through the comparison above, we decide to select the Actel FPGA IGLOO Nano for the

¹Not Available.

PAM implementation. A complete package IGLOO Nano Starter Kit including AGLN250V2 IGLOO Nano chip (as depicted in Fig.4.21) was bought, which enables us to quickly evaluate IGLOO Nano family. The kit allows developing simple designs using switches and LEDs, or by removing the jumpers to the switches and LEDs, or you can then use the board to build your full system with other devices with all users I/Os available. Then you can also measure power to the device, and to each I/O bank to evaluate the power consumption of your design (dynamic, static, and Flash*Freeze modes) with the core operating at either 1.2V or 1.5V.

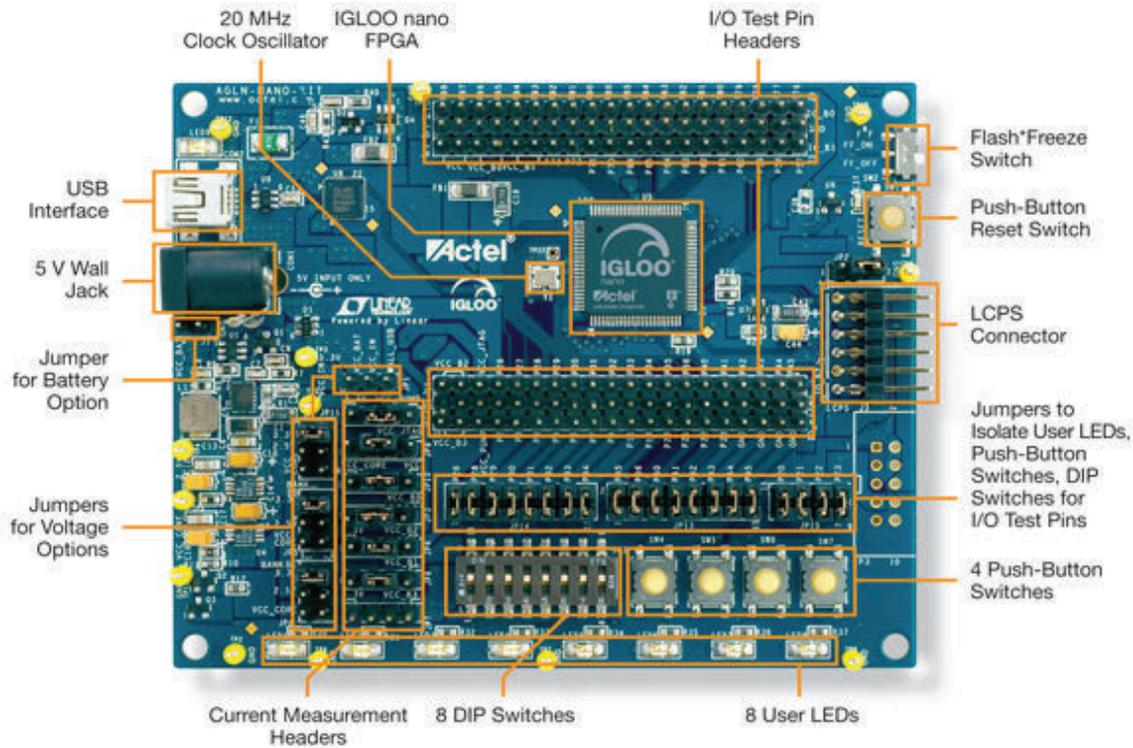


Figure 4.21: IGLOO Nano Starter Kit

To program and implement your design into the IGLOO evaluation board, the Libero Integrated Design Environment (IDE) is used. The Libero IDE and its tools enable us to create a simple design incorporating Libero IDE Catalog IP core macros, library primitives, and VHDL code. Then we can run simulation, debug and implement our program in the evaluation board using this tool. The detail description of Libero IDE tool is provided in technical document [67]. The PAM design with similar functionality as first implementation in microcontroller is created in Libero IDE as depicted in Fig.4.22.

The simple design above includes:

- A coreABC serves as simple, configurable and programmable controller toward the implementation of Advanced Microcontroller Bus Architecture (AMBA) Advanced Periph-

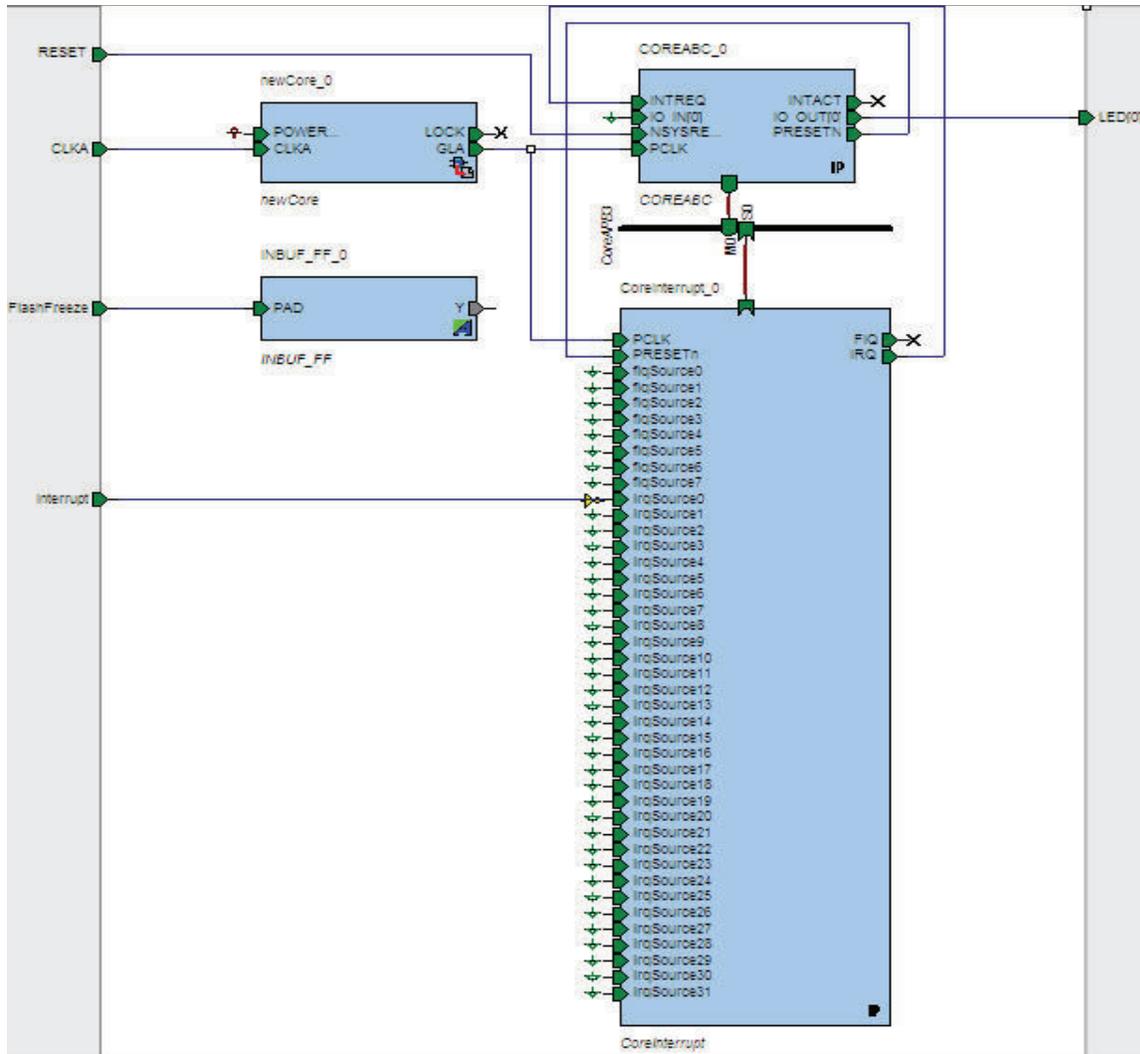


Figure 4.22: PAM design in Libero IDE after connecting components

eral Bus (APB) based designs. CoreABC supports a comprehensive assembler based configurable instruction set architecture and extensive and flexible configuration of size and feature options, allowing it to be tuned to meet the resource constraints and processing power requirements of a wide variety of applications.

- A Phase-Locked-Loop (PLL) block serves as clock generator to generate appropriate frequency for system from 20MHz clock quartz source available on the evaluation board. Here, PLL is set to provide a frequency of 4MHz to PAM system.
- An Interrupt component is used to manage all interruption sources.
- An input buffer INPUT_FF is used for Flash-Freeze mode management.

The coreABC controller connects with other components through advanced microcontroller bus architecture (AMBA) advanced peripheral bus (APB3), which serves as interconnecting an APB master and up to 16 APB slaves. After running synthesis and running program in IGLOO board, several results are summarized in Tab.4.8. Since simple application is programmed to implement PAM in AGLN250V2 IGLOO Nano chip, the resource usage is small 5% in logic cell and 12% in block RAM. As seen in this table, the power dissipation of AGLN250V2 IGLOO Nano chip is low when consuming only 1.1mW in active state, and its consumption reduces up to 26uW and 24uW in Idle and Flash-Freeze states respectively. The time-to-response to interruption arrival is 1.5us.

Table 4.8: Synthesis and measurement results of PAM implementation in FPGA IGLOO Nano

Logic Cells Usage	273 of 6144 (4%)
Block RAM usage	1 of 8 (12%)
Power consumption (Flash-Freeze)	24 μ W
Power consumption (Idle)	26 μ W
Power consumption (Active)	1.1mW
Time-To-Response	1.5 μ s

Conclusion

Based on measured results of the PAM implementation, we note that Actel FPGA IGLOO Nano family is much higher energy-efficient than microcontroller in all operating modes (low-power and active modes), when realizing the same task. That is pivotal advantage of using FPGA IGLOO Nano in wireless sensor node that has a limited energy budget. Related to time-to-response to any arrived event, microcontroller reacts slightly faster than IGLOO Nano controller. To have a clear comparison about the performance and energy-efficiency between microcontroller and FPGA IGLOO Nano, a test bench of complex task is realized in the next section.

4.4.2 Adjustable regulator implementation

Test bench realization

Applying the DVFS technique in sensor node for power management stresses the need of adjustable regulator, which regulates dynamically the supply voltage for node components based on arrived events during operation. We aim to design an adjustable regulator that has a quick switching time between voltage levels. This feature is very crucial due to its impacts in application performance. The realized test-bench includes three main components such as a microcontroller/FPGA IGLOO Nano, a digital potentiometer and a digital regulator. While the first component is used to implement the PAM that undertakes power management of our sensor node, the second one serves to adjust the output resistor value following the command from PAM. The output resistor connects to the digital regulator to provide needed supply voltage.

The AD5204 digital potentiometer as depicted in Fig.4.23 is a Dual-Line In Package (DLIP) that provides a 4-channel, 256-position digitally controlled variable resistor (VR) device. Changing the programmed VR settings is done by clocking an 11-bit serial data-word into the SDI pin as shown in Tab.4.9. The first three address bits MSB (Bit 10 through Bit 8) of this data-word indicate the channel address, and the last eight bits LSB (Bit 7 through Bit 0) provide the data value to compute the output resistor as explained afterwards.

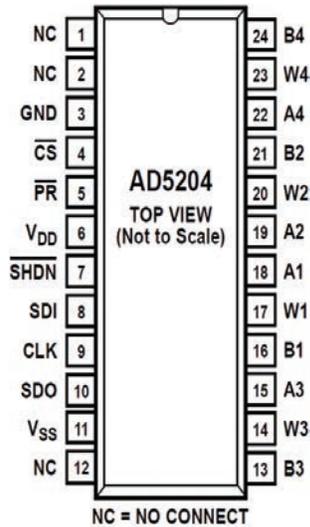


Figure 4.23: AD5204 digital potentiometer pin configuration

Table 4.9: Serial-Data Word Format

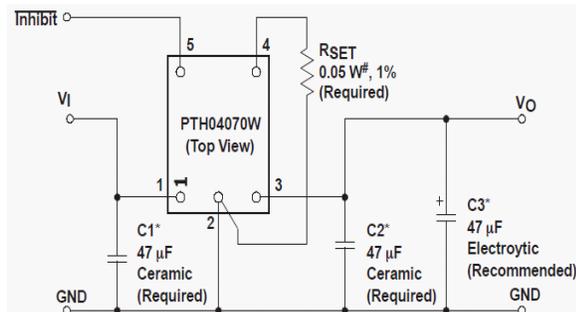
Address			Data							
B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
MSB		LSB	MSB							LSB
2^{10}		2^8	2^7							2^0

The nominal resistance between the Terminal A and Terminal B is 10kOhms. The output resistance (V_{WB}) has 256 points accessed by the Wiper terminal (W). The last 8-bit data-word is decoded to select one of the 256 possible settings. The first connection of the wiper (W) starts at Terminal B for the 0x00 data that corresponds to default resistor value $R_{WB} = 45\Omega = RW$. Each LSB data value increase moves the wiper up the resistor ladder until the last output resistor value is reached at 10000 Ω . The wiper W does not directly connect to Terminal A. The transfer equation determining the digitally resistor value between the Wiper W terminal and B terminal is follows:

$$R_{WB} = \frac{Dx}{256} \times R_{AB} + R_W \quad (4.14)$$

Where Dx is the data contained in the last 8-bit LSB data-word, and $R_{AB} = 10k\Omega$ is the nominal end-to-end resistance. The other feature specifications of AD5204 digital potentiometer is described in [68].

The PTH04070W digital regulator as depicted in Fig.4.24 is a highly integrated, low-cost switching regulator module that delivers up to 3 A of output current. Due to its high efficiency and low-power dissipation at current generation, small size and low cost, this device is attractive for a variety of applications. The operation input voltage is 3.3V or 5V when the second one is selected in our study to generate output voltage over the range from 0.9 V to 3.6 V. Based on the utilization of single external resistor R_{SET} , this regulator may generate needed supply voltage as depicted in Fig.4.24. In our case study, the two supply voltage values required are 3V and 3.6V, which correspond to two values $1k\Omega$ and 60Ω of R_{SET} .



V_O Req'd (V)	R_{SET} (k Ω)	V_O Req'd (V)	R_{SET} (k Ω)	V_O Req'd (V)	R_{SET} (k Ω)
0.900	Open	1.475	12.3	2.55	2.16
0.925	353	1.50	11.6	2.60	2.00
0.950	175	1.55	10.5	2.65	1.85
0.975	116	1.60	9.49	2.70	1.71
1.000	85.9	1.65	8.64	2.75	1.58
1.025	68.0	1.70	7.90	2.80	1.45
1.050	56.2	1.75	7.24	2.85	1.33
1.075	47.7	1.80	6.66	2.90	1.22
1.100	41.3	1.85	6.14	2.95	1.11
1.125	36.4	1.90	5.67	3.00	1.00
1.150	32.4	1.95	5.25	3.05	0.904
1.175	29.2	2.00	4.86	3.10	0.810
1.200	26.5	2.05	4.51	3.15	0.720
1.225	24.2	2.10	4.19	3.20	0.634
1.250	22.2	2.15	3.89	3.25	0.551
1.275	20.5	2.20	3.61	3.30	0.473
1.300	19.0	2.25	3.36	3.35	0.397
1.325	17.7	2.30	3.12	3.40	0.324
1.350	16.6	2.35	2.90	3.45	0.254
1.375	15.5	2.40	2.70	3.50	0.187
1.400	14.6	2.45	2.51	3.55	0.122
1.425	13.7	2.50	2.33	3.60	0.060
1.450	13.0				

Figure 4.24: Characteristics of PTH04070W digital regulator

The principle of block interconnection in the test-bench is shown in Fig.4.25. For this experiment of supply voltage adjustment, generated event interrupt is corresponding to urgent data arrival (as explained in Section 4.2.1). As depicted in this figure, when an event interrupt is generated, the PAM wakes up to execute the ISR instruction code, which writes two 8-bit data word to digital potentiometer register through Serial Peripheral Interface (SPI) protocol. While

the first 8-bit data word contains the address of output resistance channel, for example if R_{WB0} is used, the 8-bit data address is 0x00 in hexadecimal, then the second 8-bit data word includes the value to compute required output resistance R_{WBx} . When the new R_{WBx} value is set, the digital regulator switch the output voltage to new value with switching frequency 700KHz. As our WSN applications requires output voltage on/off control, the PTH04070W regulator incorporates an output on/off *Inhibit* control. The inhibit feature can be used wherever there is a requirement for the output voltage from the regulator to be turned off. As explained in technical document, the *Inhibit* control has its own internal pull-up to V_{IN} potential, thus an open-collector or open-drain circuit is recommended to control this input. In our test bench, the PAM can control voltage on/off option of the regulator via output *Inhibit* pin as depicted in Fig.4.25. This output pin connects to the PTH04070W regulator through BSS138 CMOS transistor to create an open-collector circuit.

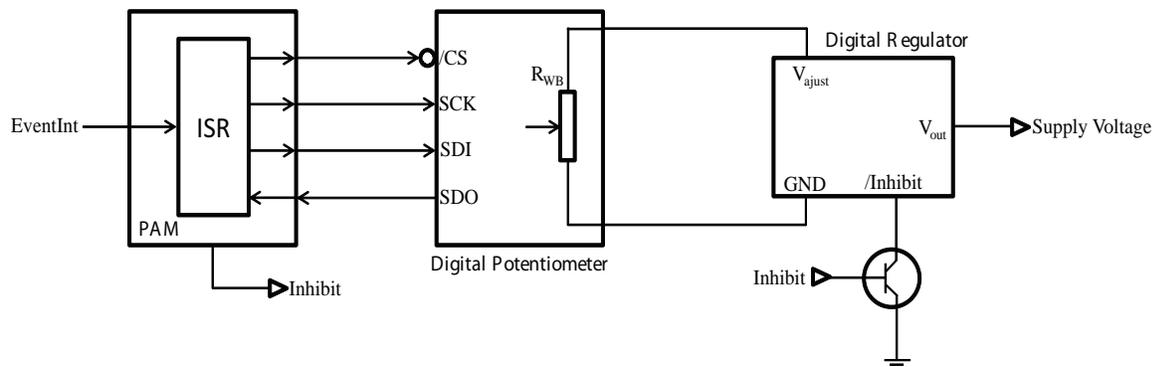
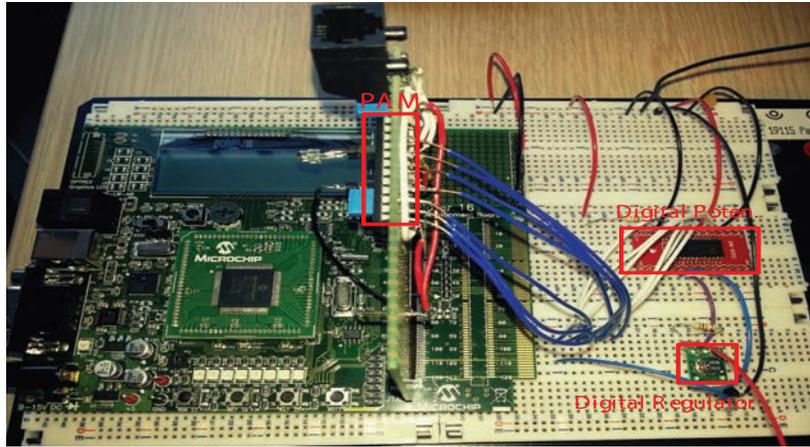


Figure 4.25: Block interconnection diagram for adjustable regulator circuit

The regulator module functions normally when its */Inhibit* pin is left open-circuit providing a regulated output whenever a valid source voltage is connected to V_{in} with respect to GND. That corresponds to BSS138 transistor to be switched off when the output *Inhibit* pin of the PAM is set to 0. Turning BSS138 transistor on (the output *Inhibit* pin of the PAM is set to 1) applies a low voltage to the */Inhibit* control pin, and disables the output of the module. Next, we realize two test-benches, the first one implements PAM in PIC24F16KA102 16-bit flash microcontroller and the second one uses AGLN250V2 IGLOO Nano chip for PAM implementation. Our objective is to measure the power consumption and overall time of voltage switching of the realized regulator test-benches. The Fig.4.26.a depicts the first realized test-bench in which the PAM, digital potentiometer and digital regulator are marked by red rectangles.

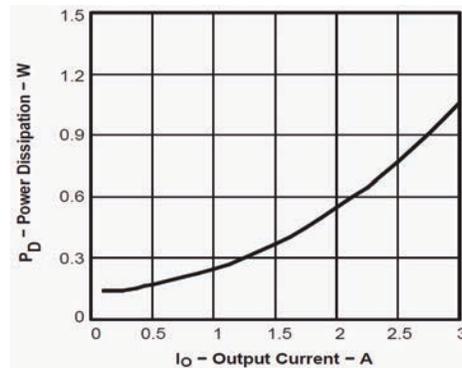
Since the PAM microcontroller power consumption is nearly similar to previous test, we focus on those of digital potentiometer and digital regulator. The Fig.4.26.b et Fig.4.26.c show their measured consumption, while the consumed power of potentiometer is stable and very small (27 μ W), the regulator dissipation is proportional to the output current. Since the node components like processor, memory and radio module require the supply current over range from 35mA to 250mA, hence the regulator dissipation is about 250mW.



a. Adjustable regulator test-bench with PAM implementation in microcontroller

	Measurement
Supply voltage	3.3V
Operating frequency	4MHz
Consumed current	8uA
Dissipated power	27uW

b. Digital potentiometer consumption



c. Digital regulator consumption versus current

Figure 4.26: Adjustable regulator test-bench with PAM implementation in microcontroller

At the time analysis of voltage switching, the overall time is computed from the moment of event interrupt arrival to the moment the new output voltage is generated. This time is given as follows:

$$T_{Switch} = T_{PAMWakeUp} + T_{ResSet} + T_{VolSet} = \frac{3}{4MHz} + \frac{16}{4MHz} + \frac{1}{700KHz} = 6.2\mu s \quad (4.15)$$

The small switching time of our regulator avoids negative impact on application performance, which is appropriate to be applied in wireless sensor node.

In the second test bench for adjustable regulator, the PAM is implemented in AGLN250V2 IGLOO Nano chip. Thanks to its flexibility, the programmers can reprogram this FPGA IGLOO Nano chip to do any required tasks by using logic blocks and logic gates available on it. The

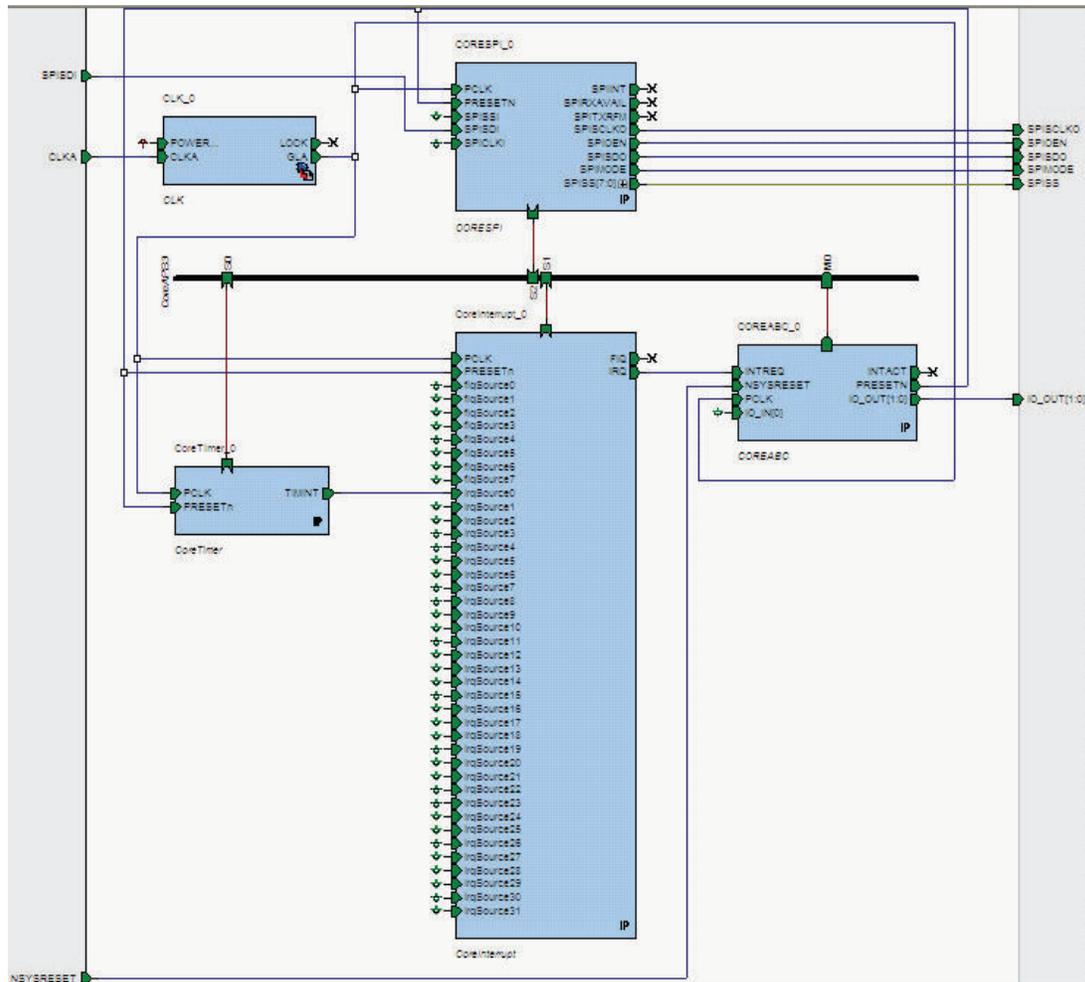


Figure 4.27: Diagram of PAM block implemented in IGLOO Nano for adjustable regulator

Fig.4.27 depicts the interconnection between the block to realize PAM functionality for adjustable regulator when implementing it in AGLN250V2 IGLOO Nano chip. As shown in this figure, two components are added like SPI block and timer block. While the first block serve as serial communication between PAM and digital potentiometer to adjust the output resistance value, the second block is used to periodically generate interruption events for voltage switching. Next, we assign the input/output pins on the FPGA IGLOO evaluation board that link to the input/output gates as depicted in Fig.4.27.

The Fig.4.28 presents the realized test bench for adjustable regulator using the AGLN250V2 IGLOO Nano chip to implement our PAM. The RTL synthesis is run to compute logic resources required to realize PAM operation. Then the program code of PAM functionality is loaded into the AGLN250V2 IGLOO Nano chip to run application. The synthesis and power consumption results are given in Tab.4.10. As seen in these results, the logic resources required to realize the PAM functionality significantly increase as 40% in logic cell usage and 12% block RAM

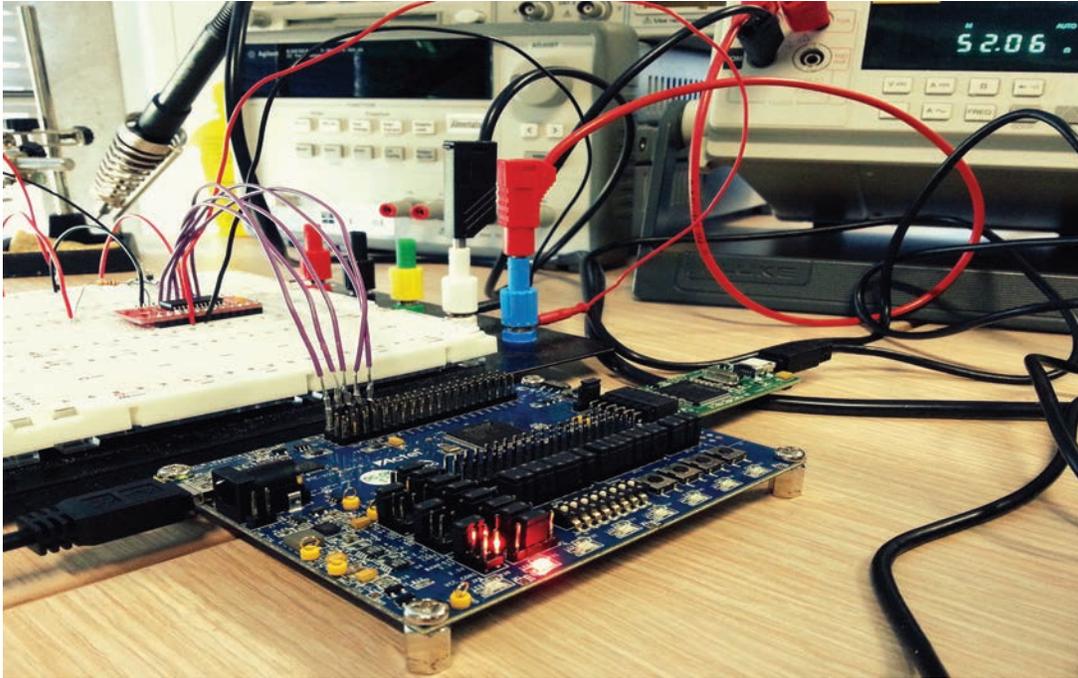


Figure 4.28: Adjustable regulator test-bench with PAM implementation in IGLOO Nano

usage. Related to power consumption, the PAM dissipates about $28\mu\text{W}$ in idle state, however PAM consumes 3.2mW in active state (up to three times more than its consumption in previous test 4.8) due to additional logic resources. The required time for voltage switching in this case is about $20\mu\text{s}$, which is three times more than required switching time when implementing PAM in microcontroller.

Table 4.10: Adjustable regulator test-bench results when implementing PAM in FPGA IGLOO Nano

Logic Cells Usage	2461 of 6144 (40%)
Block RAM usage	1 of 8 (12%)
Power consumption (Flash-Freeze)	$24\mu\text{W}$
Power consumption (Idle)	$28\mu\text{W}$
Power consumption (Active)	3.2mW
Time-To-Response	$20\mu\text{s}$

Conclusion

The use of FPGA IGLOO Nano to implement the PAM for realizing any complex task is more advantageous than implementing the PAM in microcontroller in term of energy. But we note that the more complex task PAM executes, the more energy the PAM dissipates when being

implemented in IGLOO Nano chip due to additionally required logic sources. At the term of processing time, the microcontroller offers processing capability faster than the AGLN250V2 IGLOO Nano chip ($0.75\mu s$ compared to $20\mu s$). Based on the obtained results of the PAM implementation in microcontroller and FPGA IGLOO Nano, a synthesis including the advantages and disadvantages of each material solution is given in following Tab.4.11.

Table 4.11: Synthesis of each material solution for PAM implementation

	Energy-Efficiency	Performance	Implementation complexity	Price
Microcontroller	+	+++	++	+
FPGA IGLOO Nano	++++	-	-	-

According to the synthesis above, the use of FPGA in the node architecture does not present immediate gain, particularly in term of time-to-response. However, it should be noted that the proposed solution of PAM implementation in AGLN250V2 IGLOO Nano chip as depicted in Fig.4.27, is just a prototype in which our PAM is implemented in FPGA soft-cores. As described in Fig.4.27, the soft-cores such as coreABC, coreSPI, coreTimer, coreInterrupt communicate to each other through external bus (coreAPB3), therefore the communication time is very high that increases the time-to-response. To avoid this drawback, the PAM implementation in the FPGA can be realized under the form of Finite State Machine (FSM), which eliminates the use of several soft-cores and external bus. That not only improves the time-to-response, but also decreases the used logic resource that reduces the power consumption. This optimization could be realized in the future work in the industrial process. Another reason to consider the use of FPGA in our study is that the FPGA device may be reprogrammed to do any task, which is fitted to its number of gates and available logic resource. Vice versa, microcontrollers already have their own circuitry and instruction set that the programmers must follow in code programming, which restricts it to be used in very complex application of wireless sensor network.

4.4.3 Hardware failure detection implementation

In this part, the platform of hardware failure detection based on online current measurement is introduced. The realization of this platform aims to measure the power consumption, measurement resolution and time-to-response against hardware failure occurrence, in order to show the feasibility of our approach. That allows sensor node autonomously to detect a hardware problem occurred inside, and then launches functioning self-test to localize precisely the failed component. The PROTEUS ISIS soft-tool is used to design the interconnection between the components of the platform as shown in Fig.4.29.

Based on the principle of designed component interconnection in PROTEUS ISIS tool, the schema of Printed Circuit Board (PCB) is generated by using PROTEUS ARES soft tool. This tool allows the designers to place the components in appropriate locations in the PCB board, and then to make trace connections between these components as depicted in Fig.4.30. The realized PCB includes the following components:

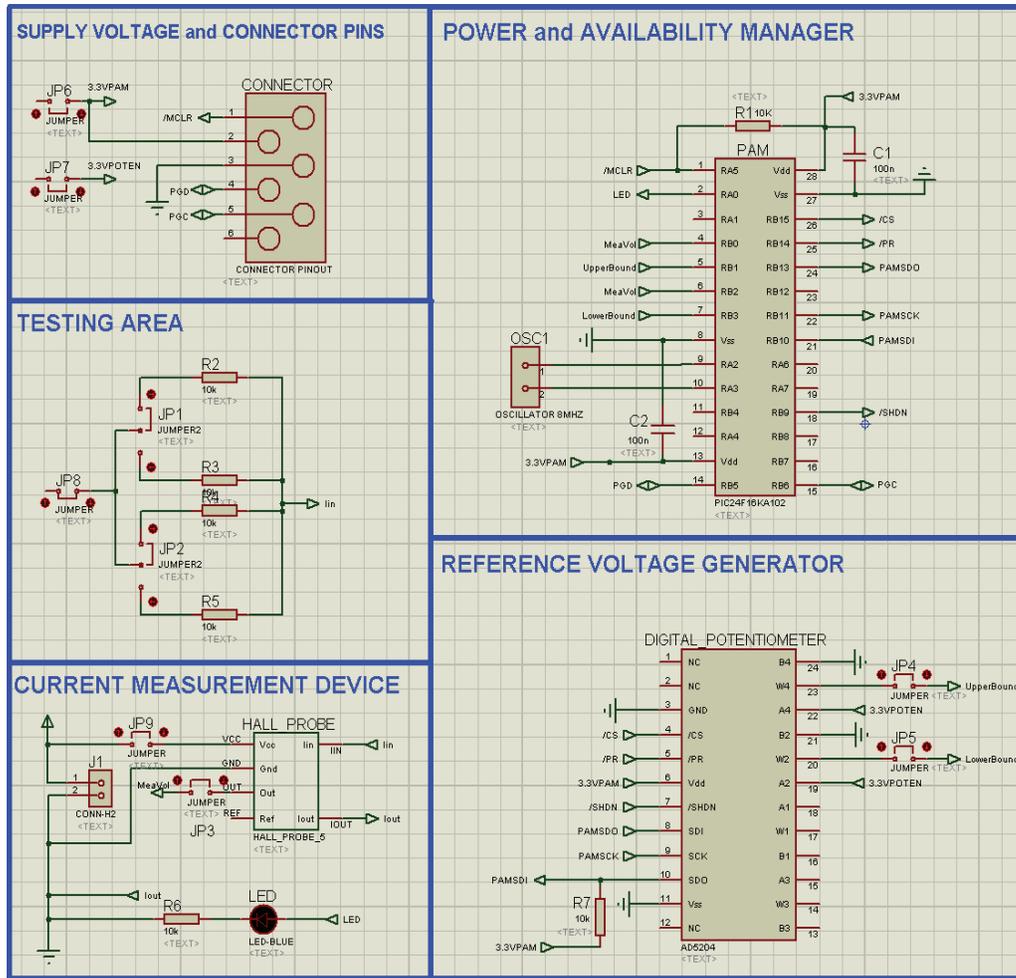


Figure 4.29: Block Interconnection of hardware failure detection platform

- A low-power microcontroller Microhip PIC24F16KA102 is used to implement the Power Availability Manager (PAM).
- An ultra-low power digital potentiometer AD5204 is used to generate external adjustable voltage references, which are connected to the inputs of analog comparators of the PAM for voltage comparisons.
- A HALL probe is used to measure current consumption and convert this current in voltage, which is connected to the inputs of analog comparators of the PAM for voltage comparisons.
- A testing zone where the resistances are placed to simulate the power consumption of each node component (processor, ram memory, sensor or radio module).
- A led for indication of hardware failure detection and the jumpers for consumption mea-

surement.

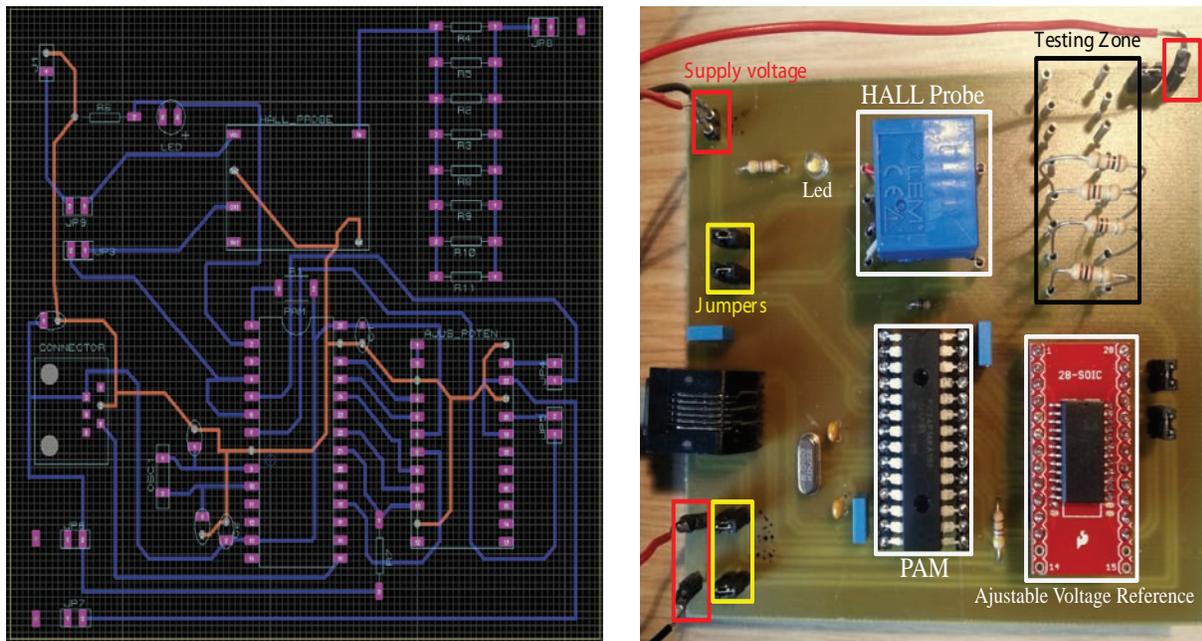


Figure 4.30: Hardware failure detection PCB

The connection between the components is shown in Fig.4.31, in which the program is loaded into the PAM from computer. Initially, the PAM communicate with the digital potentiometer AD5204 through SPI interface to set two voltages $V_{lowerBound}$ and $V_{upperBound}$. These threshold values can be adjusted dynamically during node operation as explained in [68]. When comparing to the converted voltage outputted from Hall probe, if this voltage is inferior to $V_{lowerBound}$ or superior to $V_{upperBound}$, an interruption is generated that releases PAM from sleep mode. Then PAM executes a piece of code, called the Interrupt Service Routine (ISR), directly associated with this interruption, to provide the required response program. In our test, the led is blinked when a hardware failure is detected.

The values of power consumption are shown in Tab.4.12. Since the PAM spends most of its lifetime in sleep mode, our method introduces hardware failure detection design with low-power consumption (about 73mW). This approach increases only 6.5% the power consumption of sensor node, which is used in the hazardous gas detection application as previously described.

Based on the technical document and demonstration, our approach is able to detect a minimum change of 100mA of measured current, which is suitable to apply in most of WSN applications such as hazardous gas detection, anti-intrusion [Nicolas Ferry, 2011 [26]], etc. Additionally, the latency for the PAM to pass from sleep mode to active mode is about $3T_{clk}$. In our test, with the operating frequency of 4MHz, time-to-response against hardware failure occurrence

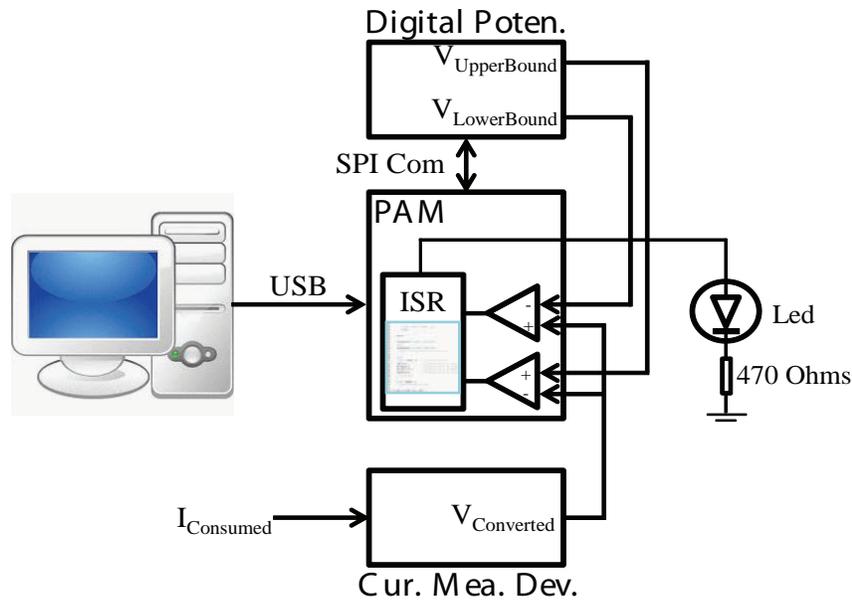


Figure 4.31: Principle of self diagnosis implementation

Table 4.12: Measurement of power consumption

Component	Mode	Vol. (V)	Cur. (mA)	Pow. (mW)
PAM	Sleep	3.3	0.05	0.17
	Active	3.3	4.9	16.2
	Active (led)	3.3	5.4	17.9
Cur. Mea. Dev.	Active	5	14.5	72.5
Digital Poten.	Active	3.3	0.008	0.03

is about 0,75us. The significant results in implementation test show interest and feasibility of hardware failure detection technique for wireless sensor node.

Considering economic aspect, the usual solution of the enterprises is to double the number of sensor nodes in the network, in order to deal with hardware-faulty nodes. However, it is uncertainly that these redundant nodes operate when is necessary, when they are in inactive mode for a long time in harsh environment. This risk is a waste of money in term of WSN cost and maintenance. Meanwhile, our approach highly enhances the node availability as shown in simulation, hence increases slightly WSN cost and decreases WSN maintenance cost. This is an important advantage of our approach.

4.5 Conclusion

In this chapter, we have presented a novel design that takes both energy and availability constraints into consideration, in order to increase energy-efficiency, and improve availability of wireless sensor node. This design approach includes a smart Power and Availability Manager (PAM) device. By applying the DPM and DVFS techniques, the PAM can control the effective use of energy based on arrived discrete events. Also, the PAM supervises sensor node functionality to detect any hardware failure based on accurately online measurement of supply current. The extra components like processor, memory, sensors are added in sensor node to replace the failed component if detected, which leads to increase node availability.

To provide a relevant characterization of energy gain, we have considered the execution of different scenario tests of hazardous gas detection application in CAPNET-PE tool. The strategy of applying both DPM and DVFS techniques provides significant energy gains in all cases, ranging from 72% when event rate is low (24 events/days) to 35% when event rate is high (240 events/day). Meanwhile, the energy overhead of our approach is about 0.01% over total energy consumption of sensor node. Thus, our approach is suitable to be applied in battery-powered device like wireless sensor node. We have also run the availability analysis by using SPNP tool to show the effect of our reconfiguration approach on the availability improvement of wireless sensor node. When fixing the availability threshold at 90%, the simulation results demonstrate that our approach significantly increases the node availability from 273 days to 3.4 years (up to 453%). As a result, the cluster availability increases significantly from 238 *days* (without reconfiguration) to 3.35 *years* (with reconfiguration), which is up to 513%. We conclude that our reconfiguration approach improves highly the node availability that leads to increase highly cluster availability.

Afterwards, to demonstrate the feasibility of our approach, several implementations in real materials were made. Firstly, the PAM was implemented by using two low-cost, low-power, small size and high security material solutions: Microchip 16-bit microcontroller and FPGA IGLOO Nano. The FPGA was considered in our study to provide more material solution in PAM implementation next to traditional microcontroller device. Thank to its high flexibility, the FPGA device may be reprogrammed to do any task that is fitted to the number of gates available in this FPGA device. Vice versa, microcontrollers already have their own circuitry and instruction set that the programmers must follow in code programming, which restricts it to certain tasks. Besides, other improvements in term of increasingly better power efficiency and decreasing prices make FPGA rule almost electronic embedded systems in the near future. Through experimentation results, we noted that the FPGA IGLOO Nano is much higher energy-efficient than microcontroller for PAM implementation to realize any complex task. That is pivotal advantage of using FPGA IGLOO Nano in wireless sensor node that has a limited energy budget. However, at the term of processing time, the microcontroller offers processing capability faster than the FPGA IGLOO Nano chip. For example, in the adjustable regulator implementation, the power consumptions of the PAM in active and sleep modes are 3.2mW and

$28\mu\text{W}$ when implementing it in FPGA IGLOO Nano, which are much less than the power consumptions of the PAM in active and sleep modes (17.9mW and $170\mu\text{W}$) when implementing it in Microchip 16-bits microcontroller. However, at the term of processing time, the microcontroller offers processing capability faster than the AGLN250V2 IGLOO Nano chip ($0.75\mu\text{s}$ compared to $20\mu\text{s}$). Therefore, based on the requirement of each application, the FPGA IGLOO Nano may be targeted to energy-critical applications, and the microcontroller may be targeted to time-critical application. It should be noted that our proposed solution of PAM implementation in AGLN250V2 IGLOO Nano chip as depicted in Fig.4.27, is just a prototype and it could be optimized in the industrial process to reduce energy consumption and processing time delay. Secondly, a test bench of hardware failure detection based on online current measurement was made to measure its energy overhead and performance. The obtained results showed the feasibility of our approach.

Chapter 5

APPROACH AT NETWORK LEVEL

As we know, Wireless Sensor Networks (WSNs) are energy and resource constrained networks, which are made up of small sensor nodes. These pervasive micro-sensing and actuation devices may revolutionize the way we understand and manage complex physical systems. The capabilities for detailed physical monitoring and manipulation offer enormous opportunities for almost every scientific discipline. The WSNs also provide ubiquitous embedded processing platform with exciting capabilities that allows remote monitoring of target environments. For this reason, WSNs are favorite candidates in various applications, even including surveillance applications in inhospitable environments such as remote geographic regions or toxic locations, sensing and maintenance in large industrial plants, seismic activity detection, medical sensing, micro-surgery, military surveillance and combat, and smart office spaces.

In practically all such applications, key requirements include scalability, robustness in regard to various disturbances and uncertainties, evolution through autonomous reconfiguration and optimal redistribution of resources. And the most important factor is wireless connectivity because for most envisioned applications, the monitoring environments do not have installed infrastructure for either communication or energy supply. Since wireless sensor nodes are battery-powered devices, we aim to select consistent communication topology to deploy WSNs in the monitoring environments. As explained in Section 2.1, cluster topology is applied in our study, where the WSN is sub-divided into groups of sensor nodes called clusters, and the nodes in same cluster send data to its own cluster-head called Gateway. Using this topology allows sensor nodes to communicate data over smaller distance in the cluster environment, which reduces their transmitting energy.

Major issue is raised to be the Gateway unavailability due to energy depletion or hardware failure when using cluster topology. That is a critical issue, because we will lose the data from all the nodes of a cluster if their Gateway ceases to operate due to energy depletion or hardware failure. That stresses the need of an appropriate algorithm to find temporary cluster-head, which can replace the Gateway during the recharge of its depleted battery or its reparation. This mechanism is very important for maintaining packet delivery ratio and data throughput in the network, while decreasing the dissipated energy for data re-transmission due to communication

rupture. In this chapter, we introduce a novel cluster-head selection approach to extend network lifetime and reliability by taking obstacle-aware criteria into consideration. Another crucial issue should be considered in network communication is Sink unavailability, because supervisors loss the connection with the whole network. This chapter also presents our solution to deal with this critical problem.

The chapter is organized as follows: Section 5.1 presents cluster structure used in our study. Since the WSN structure is various in different applications, it is necessary to specify the cluster structure that allows facilitating our work. As explained above, a cluster-head selection algorithm is indispensable when employing the cluster topology. However, several factors should be considered to propose an appropriate algorithm, and those are described in Section 5.2. Next, novel cluster-head selection algorithm is presented in Section 5.3. The Section 5.4 introduces our approach to deal with Sink unavailability issue. Then our approach is evaluated through network simulation in Section 5.5. Finally, Section 6.3.1 presents our conclusion.

5.1 Cluster Structure Extension

As explained previously in section 2.2, when using cluster topology to deploy the wireless sensor network in the environment, a major issue is raised to be the Gateway unavailability, as we will lose the data from all the nodes of a cluster if their Gateway ceases to operate due to energy depletion or hardware failure. We could increase the Gateway availability by adding redundant Gateway nodes. However, this traditional solution is very expensive in terms of battery energy and price for large-scale networks because the number of clusters is high. For this reason, we assume that the clusters are able to run their structure reconfiguration to deal with Gateway unavailability issue, which allows us to avoid adding redundant Gateway nodes.

As stated in section 2.1.2, the cluster structure and the location of all sensor nodes are fixed during network operation. Our study context is different with the context of reference works, in which the sensor nodes are distributed randomly, and the structure of clusters is changed versus time. Fig.5.1 depicts our cluster structure, four types of sensor are considered:

- Gateway or cluster head of cluster is responsible for receiving and aggregating data, then transmitting it directly to the Sink.
- Normal nodes capture and deliver data directly to the Gateway.
- Temporary Cluster-head (Tempo-CH) is similar to normal node, it will replace temporarily cluster head role if the Gateway is suddenly down due to energy depletion or hardware failure. That allows avoiding communication disruption with the base station (Sink) because the cluster-head selection process takes an amount of time. Then it undertakes selection mechanism to find future cluster head in the candidate set. The Tempo-CH node itself is also a cluster-head candidate.

- A set of cluster-head candidate nodes (CH-Can) that are selected during network deployment, these nodes are similar to normal node. Their location is much closer to the Sink than the normal nodes, because transmitting cost is more expensive when the cluster-head is far from the Sink.

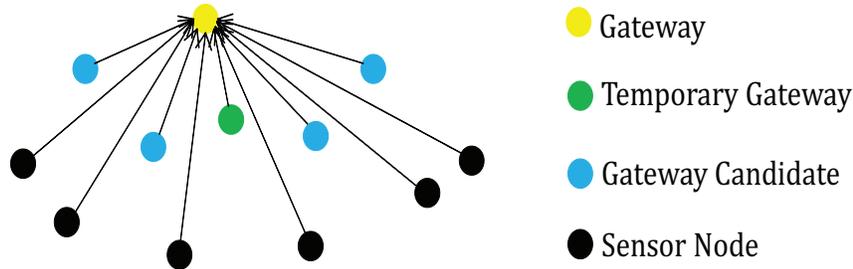


Figure 5.1: Structure of sensor cluster

5.2 Major Factors in Cluster-Head Selection Algorithm

5.2.1 Factor of residual energy

Since an embedded battery has limited capacity that restricts the service time of sensor node, most recent WSNs outperform this drawback by harvesting energy from the environment directly or recharge an energy storage system. However, this kind of energy is unstable and mainly depends in weather evolution, hence it is not trivial to estimate the node autonomy and the amount of recharging energy. To overcome this problem, including weather parameters at design stage and weather forecasts at runtime is essential for autonomy management, in which most criteria such as solar, wind, and temperature must be taken precisely into account. In previous works, a Power Estimator called CAPNET-PE [Nicolas Ferry, 2011 [26]] was built to estimate the whole node power/energy consumption based on business scenario, and predict energy harvested from the environment based on the meteorology. This tool takes the inputs such as consumption behavior of node elements, scenarios composing of rule-based events (the start time, the duration, the periodicity for periodic events, and the number of repetitions), meteorology, Dynamic Power Management (DPM) policy or Finite State Machine containing the states of system consumption and the transitions between them. Based on these data, it provides as output the consumption of each node element, and the energy scavenged from environment. CAPNET-PE can extract the weather data directly from Internet websites as meteorology data are now widely available through multiple sources. These data are then combined together to produce an energy function prediction over time. All the steps of energy estimation of CAPNET-PE tool are described in [Nicolas Ferry, 2010 [27]], and it can predict

1-day up to 3-days forward with relatively good accuracy but a complete week with more uncertainty. Using CAPNET-PE tool, we can predict precisely the average power scavenged from each external source such as solar and wind sources, and then compute their corresponding amount of harvesting energy per hour, which allows estimating the lifetime of sensor nodes during our simulation of wireless sensor network. These energy data will be reused in our approach of cluster head selection.

The residual energy is one of pivotal factor to balance the total energy dissipation among the candidate nodes for performing the cluster head selection. Initially, all the nodes are constrained with the same battery, however their remaining energy are not equal after a long running time due to two main reasons: the number of data communication, and the amount of harvesting energy that are different from one node to other node. Thereby, there are nodes with low residual energy, and other nodes having higher residual energy. If the nodes with very low remaining energy are selected as cluster heads, it can be imagined that they have a high probability to quickly run out of all their remnant energy and become dead nodes. As a result, a large number of dead nodes will appear that leads to reduce reliability and lifetime of the network. Therefore, consideration of residual energy in cluster head selection protocol has an important impact. One criteria of our approach is to avoid selecting the nodes with low remaining energy to become cluster heads. The computation of the factor of remaining energy of a node at instance t is given as follows:

$$E_{Residual}(t) = E_{InitBat} - E_{dissipate,0 \rightarrow t} + E_{harvest,0 \rightarrow t} \quad (5.1)$$

Where $E_{InitBat}$ is the initial energy of sensor node, $E_{dissipate,0 \rightarrow t}$ is the energy dissipation of node system until instance t , $E_{harvest,0 \rightarrow t}$ presents the energy scavenged from the ambient sources like solar, wind. However, to select a new cluster head, the algorithm bases not only on residual energy, but also the distance between the candidate nodes and the Sink.

5.2.2 Factor of distance to Sink

As mentioned in the section 1, the transmission energy between two sensor nodes relates to the distance between them. As described in Fig.5.2, node1 and node2 communicate with Sink node from the distances respectively d_1 and $d_2 = d_1 + \Delta_d$. Based on the Eq.1, their energy dissipation to transmit k -bit data packet are respectively as follows:

$$E_{node1 \rightarrow Sink} = E_{elec} * k + \alpha_{amp} * k * d_1^2 \quad (5.2)$$

$$E_{node2 \rightarrow Sink} = E_{elec} * k + \alpha_{amp} * k * (d_1 + \Delta_d)^2 \quad (5.3)$$

Then the disparity of transmission energy between two nodes is given as follows:

$$E_{node1 \rightarrow Sink} - E_{node2 \rightarrow Sink} = \alpha_{amp} * k * (2d_1\Delta_d + \Delta_d^2) \quad (5.4)$$

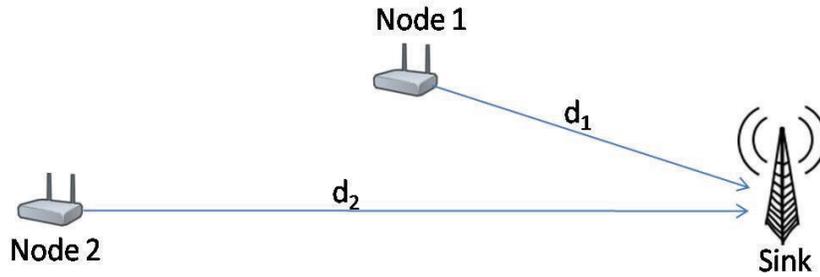


Figure 5.2: Transmission between the sensor nodes and the Sink

As seen in Eq.5.4, the larger the Δ_d is, the more energy the node 2 dissipates to communicate with the Sink compared to the node 1. As a result, the battery of the node2 is drained much faster than the battery of the node1 when they transmit the same data packet to the Sink. Therefore, the nodes located very far away from Sink node will be excluded from the set of candidate nodes for the cluster head selection. Nevertheless, when the WSNs are deployed in harsh environment containing many potential obstacles, which may interrupt the communication between the future cluster heads and the Sink, consequently lots of data packets may highly be lost. Thus, a large amount of energy is wasted due to data loss and data re-transmission. Additional parameter of the signal strength of transmission from candidate nodes to the Sink should be taken into account in cluster head selection.

5.2.3 Factor of obstacle-aware

We suggest that our network is deployed in the environment as depicted in Fig.5.3, where CH-Can1, CH-Can2, CH-Can3, CH-Can4 and Tempo-CH sensor nodes belong to the same cluster, and Gateway node is the head of their cluster. After a long running time, the battery of Gateway reaches a limited energy threshold or the Gateway suffers a hardware rupture, which does not allow it to guarantee its cluster head role, then the Gateway decides to select a temporary cluster head from the set of candidate nodes. However, some temporary obstacles may appear between the future cluster head and the Sink as depicted in Fig.5.3, this will interrupt their radio communication. For example, the obstacles can be containers or container trucks when sensor network is deployed in the seaport.

As shown in the figure above, if the CH-Can1 or CH-Can4 node is promoted to undertake cluster-head role instead of other nodes, there is highly possible to have many communication crashes with the Sink, due to the obstacle appearance between them and the Sink. As a result, the packet delivery ratio and throughput of the network are diminished. Additionally, the dissipated energy for transmission is massively increased due to data re-transmission. Thus, the factor of obstacle-aware should be considered in our cluster-head selection algorithm, which is presented in the next section.

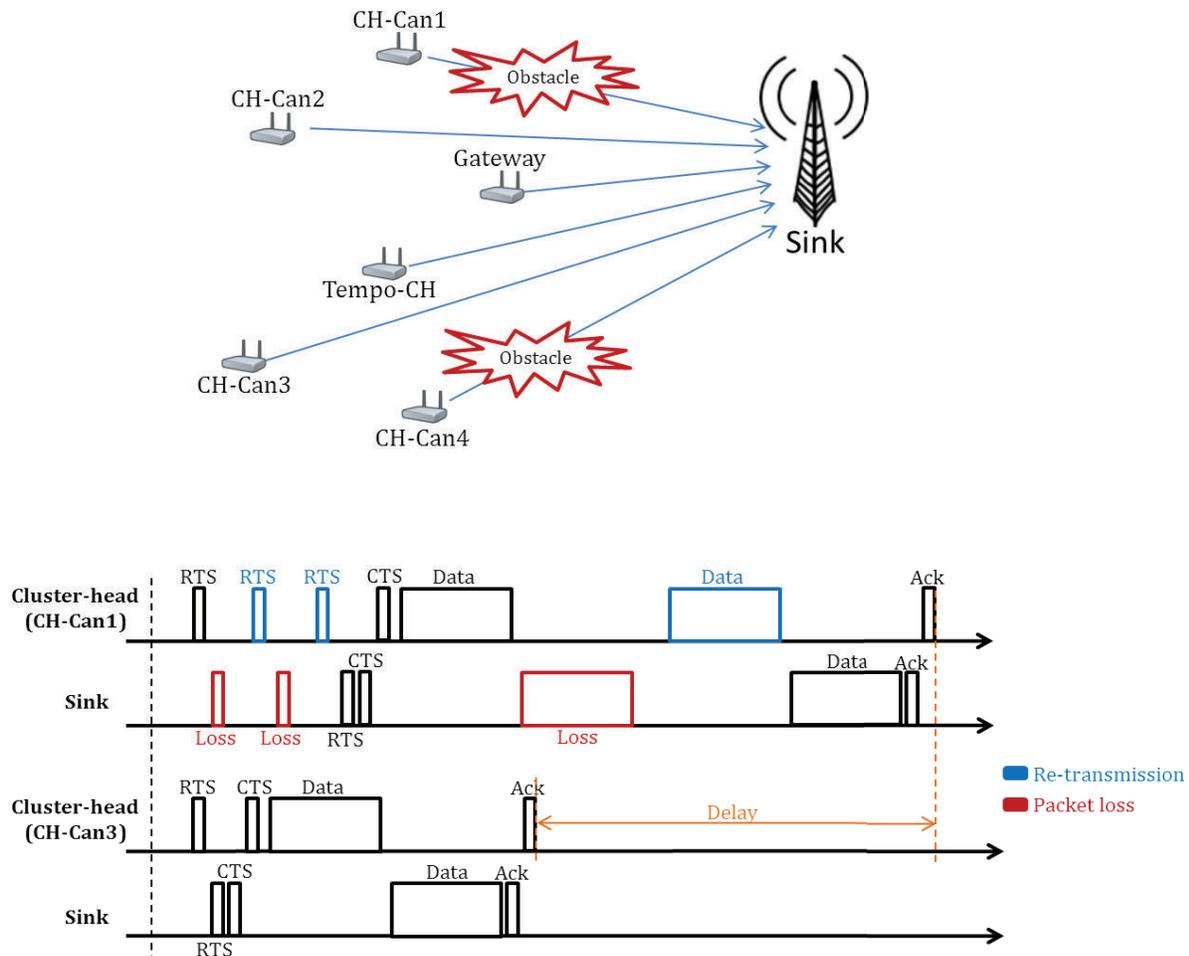


Figure 5.3: Obstacle impact in communication between cluster and Sink

5.3 Obstacle-Aware Cluster-Head Selection Approach

All the sensor nodes in the network can monitor their remaining energy by themselves. In our study, the paths between the Gateway nodes and the Sink are strictly monitored by the supervisors because they are the critical elements. We suppose there is no obstacle in these paths. However, the appearance of obstacles is possibly in the paths between the cluster-head candidates and the Sink during network operation. For example, the obstacles can be containers or container trucks when sensor network is deployed in the seaport. These obstacles may highly damage the communication link between the clusters and the Sink if these candidate nodes are promoted to become cluster-head nodes. As a result, that diminishes packet delivery ratio and throughput of the network, and increases energy dissipation due to data retransmission. Our approach has seven steps as depicted in Fig.5.4.

Step 1: When the Gateway detects its energy level lesser than an energy threshold ($E_{Gateway} <$

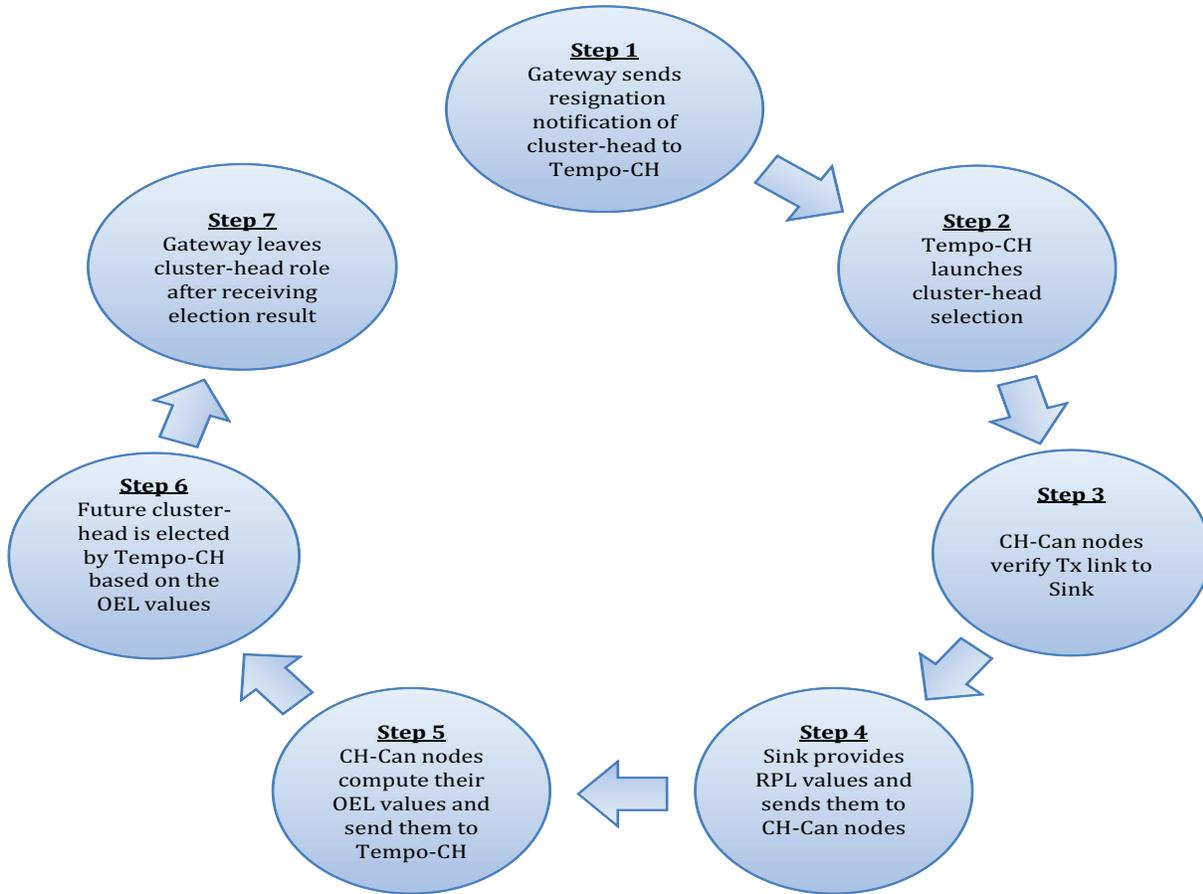


Figure 5.4: Cluster-head selection steps

$E_{lowerBound}$), it sends notification packet to Tempo-CH. This packet includes the number of transmission to Sink per hour $Nb_{TxToSink.h^{-1}}$ and the number of reception per hour $Nb_{Rx.h^{-1}}$, which are used for cluster-head selection as explained later. An appropriate threshold $E_{lowerBound}$ is selected to leave Gateway sufficient energy to operate normally such as data sensing, data transmission to cluster-head, in order to maintain the reliability of the network.

Step 2: After receiving notification from Gateway, Tempo-CH launches cluster-head selection by sending the request of operating energy level (OEL) value to all cluster-head candidates (CH-Can). The $Nb_{TxToSink.h^{-1}}$ and $Nb_{Rx.h^{-1}}$ values are included in sending packets.

Step 3: After receiving OEL request from Tempo-CH, CH-Can node saves the values of $Nb_{TxToSink.h^{-1}}$ and $Nb_{Rx.h^{-1}}$, and then sends a beacon signal to the Sink to verify if there is an obstacle in their path. The Tempo-CH also sends a beacon signal to the Sink.

Step 4: Based on the power strength of received beacon signal RSSI (Received Signal Strength Indicator), the Sink provides received power level (RPL) value, and sends it back to sender node (CH-Can node). The smaller RPL value is, the greater possibility an obstacle

appears in the communication path.

Step 5: After receiving the RPL value from the Sink, CH-Can node computes its OEL value following the Eq.5.5 and sends this value to the Tempo-CH node.

$$OEL = \frac{E_{residual}}{E_{TxToSink} * Nb_{TxToSink.h^{-1}} + E_{Rx} * Nb_{Rx.h^{-1}}} * RPL \quad (5.5)$$

Where $E_{TxToSink}$ is dissipated energy to transmit data packet to Sink, and E_{Rx} is receiving energy of a data packet. The computation of $E_{TxToSink}$ and E_{Rx} is defined in Eq.2.1 and Eq.4.2.

Step 6: After receiving OEL values from all CH-Can nodes, Tempo-CH selects the node with the highest OEL value to become future cluster-head. Tempo-CH also participates in this selection.

Step 7: When the selection is complete, Tempo-CH sends selection result to all the nodes in the cluster in order to update their routing table. The Gateway leaves cluster-head role after receiving this result.

After leaving cluster-head role, the Gateway still operates normally such as data sensing, data transmission to actual cluster-head, in order to maintain the reliability of the network. When recharging energy level of Gateway node reaches to 10% of battery capacity, it retakes the cluster-head role by transmitting notification message to all sensor nodes in the cluster, in order to update their routing table.

The case of failed Gateway due to hardware failure is also considered. To detect this problem, Tempo-CH sends periodically a beacon message to Gateway, if it does not receive any feedback from Gateway, Gateway is considered as failed. Tempo-CH replaces immediately cluster-head role, and then executes cluster-head selection from second step to seventh step. In this case, Gateway only rejoins in cluster operation after being repaired by technician.

As stated in first section, the battery capacity of the cluster-head candidate nodes is much smaller than Gateway node. Thus, several launching strategies of our obstacle-aware protocol (as summarized in Tab.5.1) are proposed to distribute evenly energy load among the cluster-head candidates. Besides, the obstacles appear instantaneously during network operating time, the objective of these strategies is to select the most appropriate cluster-head nodes versus time, in order to improve packet delivery ratio and network throughput, and mitigate the wasted energy.

5.4 Approach for Sink Unavailability Issue

As explained previously, Sink unavailability is the most critical issue in WSN operation, because the supervisors will completely loose connection with the whole network. To address this issue, a Relief Sink node is placed next to the Sink node, this redundant node will replace

Table 5.1: Three strategies of launching cluster head selection

Strategy	Description	Advantage	Drawbacks
Periodic	* Tempo-CH launches periodically cluster-head selection with given time period.	* Energy load is distributed evenly among cluster-head candidate nodes.	* Packet delivery ratio and network throughput are diminished, if the obstacles appear instantaneously between the Sink and cluster head. * Total energy overhead of cluster head selection launching is high.
Aperiodic	* When a cluster-head detects number of consecutively failed transmission superior than an allowable threshold, it asks Tempo-CH to launch new cluster-head selection.	* Total energy overhead of cluster head selection launching is small. * High packet delivery ratio and network throughput.	* Energy load is not distributed evenly among cluster-head candidate nodes.
Periodic +Aperiodic	* Combination of both above strategies.	* Energy load is distributed among cluster-head candidate nodes. * High packet delivery ratio and network throughput.	* Total energy overhead of cluster head selection launching is high.

the Sink in case of its out-of-control. Two main reasons that make the Sink stop operating are energy depletion and hardware rupture. To deal with the first case, the residual battery energy of the Sink is supervised by itself. When the residual energy of the Sink is detected to be lower than a lower bound, which is set at the network deployment, the Sink will send a notification message to Relief Sink node and wait for its activation (as depicted in Fig.5.5). After receiving the acknowledgement message of the Relief Sink for its activation, the Sink leaves its base station role and enters in recharging state. When recharging battery level is up to an upper bound, the Sink sends a message to the Relief Sink to notify that it will retake base station role back. After receiving the acceptance message of the Relief Sink, the Sink enters in active state to undertake the base station role. The Relief Sink node is back to initial state as Sleep state.

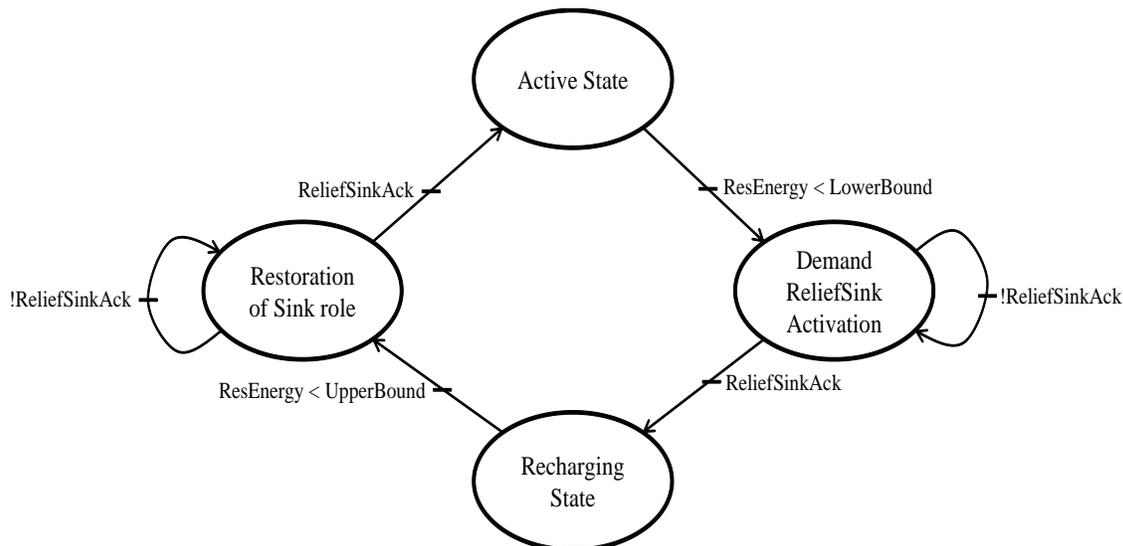


Figure 5.5: Approach principle deals with energy depletion issue of the Sink

To deal with the Sink unavailability due to hardware rupture, the supervisors is assumed to monitor functioning state of the Sink. If the Sink is instantaneously down, the supervisors detect this problem immediately because they lose the connection with the network. Afterwards, the supervisors will send a message to the Relief Sink node to activate it (as depicted in Fig.5.6). The Relief Sink is then woken up to undertake the base station role of the network. The Sink node will be repaired by a technician.

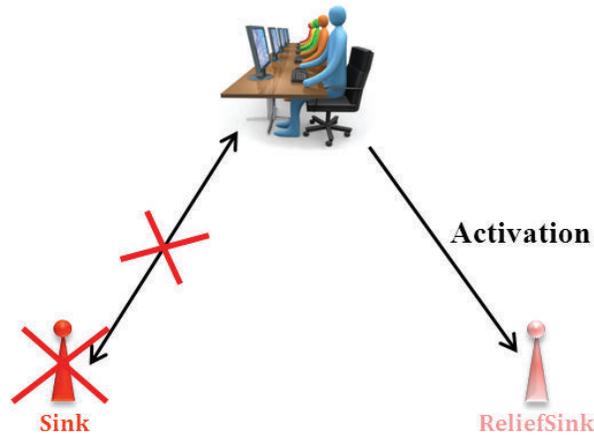


Figure 5.6: Approach principle deals with hardware rupture issue of the Sink

5.5 Evaluation

5.5.1 Simulation tool

Finding an appropriate wireless network simulator is an important part of our work since WSN test bench is costly to build and restricted in terms of flexibility. Network simulators provide us with a virtual and reproducible environment where we can design, configure, run simulation and analyze the results of different kinds of networks. Murat M.K. made a survey [M.M. Koksak, 2008 [64]] find a wireless network simulator that provides a good balance between features, efficiency, extendibility, accuracy, and easiness of use. Therefore, features, advantages/shortcomings and structure of different simulators are investigated. This survey is based on a collection of existing academic papers related to different network simulators according to defined criteria, and some of which analyze simulators' conformance to a specific project or area. Through this survey paper, the OMNeT++ tool developed by Andrea Vargas from Technical University of Budapest is selected for our network simulation due to its feature richness, powerful and accurate simulation results for large scale network. This tool is currently gaining widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community. Several main features of this tool are described in the section remainder.

OMNeT++ IDE is a discrete event simulation environment, and based on the Eclipse platform that is a very flexible system. You can move, resize, hide and show various panels, editors and navigators. One of major feature of OMNeT++ simulator is to let users describe the structure of simulation model in the NED (Network Description) language. The NED language allows users to declare their own simple modules that are inherently reusable. The Fig.5.7 shows NED description of a FIFO memory whose parameters, signals used for statistical results, and in/out gates are defined. After being saving in project library, this FIFO module can be reused.

```

simple Fifo
{
    parameters:
        volatile double serviceTime @unit(s);
        @display("i=block/queue;q=queue");
        @signal[qLen] (type="int");
        @signal[busy] (type="bool");
        @signal[queueingTime] (type="simtime_t");
        @statistic[qLen] (title="queue length";record=vector,timeavg,max;interpolationmode=sample-hold);
        @statistic[busy] (title="server busy state";record=vector?,timeavg;interpolationmode=sample-hold);
        @statistic[queueingTime] (title="queueing time at dequeue";unit=s;record=vector,mean,max;interpolationmode=none);
    gates:
        input in;
        output out;
}

```

Figure 5.7: NED description of FIFO module

The NED may also break down any complex module into smaller modules, and used as a compound module. The existing simple modules can be added in the compound module to create a simulation model like a network. That allows users to create a hierarchical structure of any complex system model. The Fig.5.8 depicts a simple network model including three components: a *gen* serves as data sender, a *fifo* memory, and a *sink* serves as base station. These components are defined as sub-modules of the compound module *FifoNet*, and the communication between them is defined as wired connection. The *FifoNet* model may be used for simulation.

After defining the module structure including parameters, signals, statistical results, and connection gates, its behavior is defined. One of the most important factors in any simulator is the programming language. The common languages used in OMNeT++ tool are C/C++ based. Thus, the module behavior is programmed by the users as C++ code. Since OMNeT++ provides an open and large Application Programming Interface (API), which allows the users to program the behavior of their own module (simple or compound) based on basic library C++ class sources. For example, a simple module is nothing more than a C++ class which has to be subclassed from basic `cSimpleModule`, with one or more virtual member functions redefined to define its behavior. The defined module class has to be registered with OMNeT++ via the `Define_Module()` macro in `.cc` or `.cpp` files and not header file (`.h`) (as depicted in Fig.5.9), in order that the compiler generates code from it.

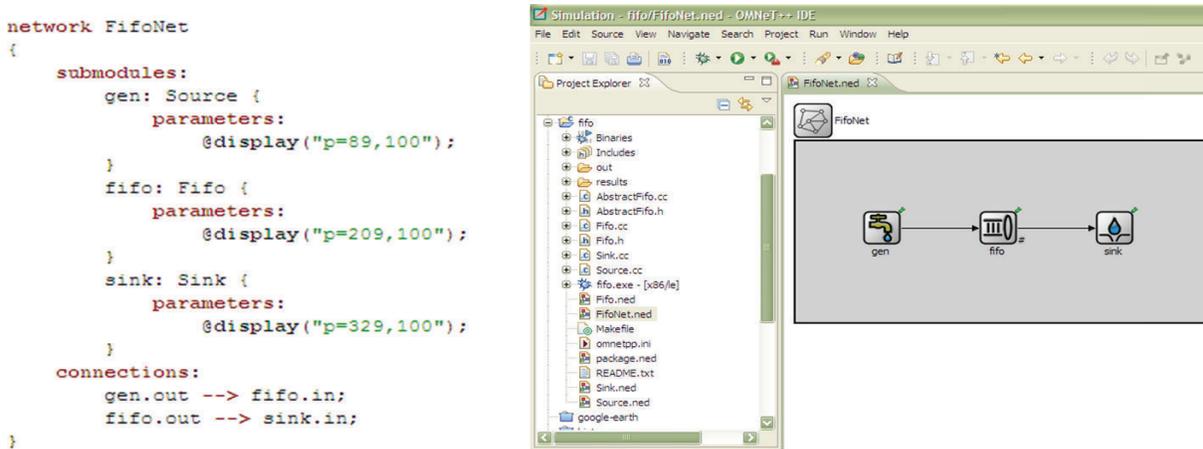


Figure 5.8: NED description of a compound module

Graphical interface is usually a user friendly option in any network simulation tool, which allows the inside of model to be seen by the user. The interaction of the user interface and the simulation kernel is realized through a well-defined interface. The user would test and debug the simulation with a powerful graphical user interface. Tkenv available in OMNeT++ tool is a portable graphical windowing user interface, which supports tracing, debugging, and simulation execution. Tkenv has the ability to provide a detailed picture of the state of the simulation at any point during the execution. This feature makes Tkenv a good candidate in the development stage of a simulation or for presentations. Important features in Tkenv are followings:

- Window separation for each module's text output.
- Scheduled messages can be watched in a window during simulation evolution.
- Event-by-event execution.
- Execution animation.
- Inspector windows to examine and alter objects and variables in the model.
- Graphical display of simulation results during execution. Results can be displayed as histograms or time-series diagrams.

A snapshot of a Tkenv interface during simulation of wireless sensor network is shown in Fig.5.10.

5.5.2 Network simulation model

To demonstrate the impact of our obstacle-aware cluster-head selection in network reliability and lifetime, four simulation cases are realized:

The image shows two side-by-side screenshots of a code editor. The left window, titled 'sensor_node.h', displays the header file. It includes preprocessor directives for 'SENSOR_NODE_H', includes 'omnetpp.h', 'string.h', and 'sucres_packet_m.h'. It defines a namespace 'dac_simu_energy_distance_qos' and a class 'sensor_node' that inherits from 'cSimpleModule'. The class has a private section with various double and int parameters for timing and transmission, and a comment for local signal results. The right window, titled 'sensor_node.cc', shows the implementation file. It includes 'sensor_node.h', defines the namespace, and implements the 'Define_Module' function. It also shows the constructor and destructor for 'sensor_node', with the constructor initializing several variables to NULL and the destructor calling 'cancelAndDelete' on those variables.

Figure 5.9: C++ codes (.h and .cpp) for programming sensor node behavior

- *Non Obstacle-Aware Cluster-Head (NoOA-CH)*: selects future cluster-head based on residual energy, and distance from cluster-head candidates to the Sink without considering the appearance of obstacles in their path. The cluster-head selection is launched periodically with given time period.
- *Periodic Obstacle-Aware Cluster-Head (P-OACH)*: adds the obstacle-aware criteria in cluster-head selection algorithm. The cluster-head selection is launched periodically with given time period.
- *Aperiodic Obstacle-Aware Cluster-Head (A-OACH)*: adds the obstacle-aware criteria in cluster-head selection algorithm. The cluster-head selection is launched when actual cluster-head detects number of consecutively failed transmission superior than a given threshold.
- *Periodic Aperiodic Obstacle-Aware Cluster-Head (PA-OACH)*: is combination of P-OACH and PA-OACH.

The description of the last three cases are presented in Tab.5.1. The network architecture for simulation is specified and configured in OMNeT++ tool as follows:

- The network consists of one cluster of 16 sensor nodes, including one Gateway, one Tempo-CH, four cluster-head candidates (CH-Can1, CH-Can2, CH-Can3, CH-Can4), and ten normal nodes (SoNo0...SoNo9).

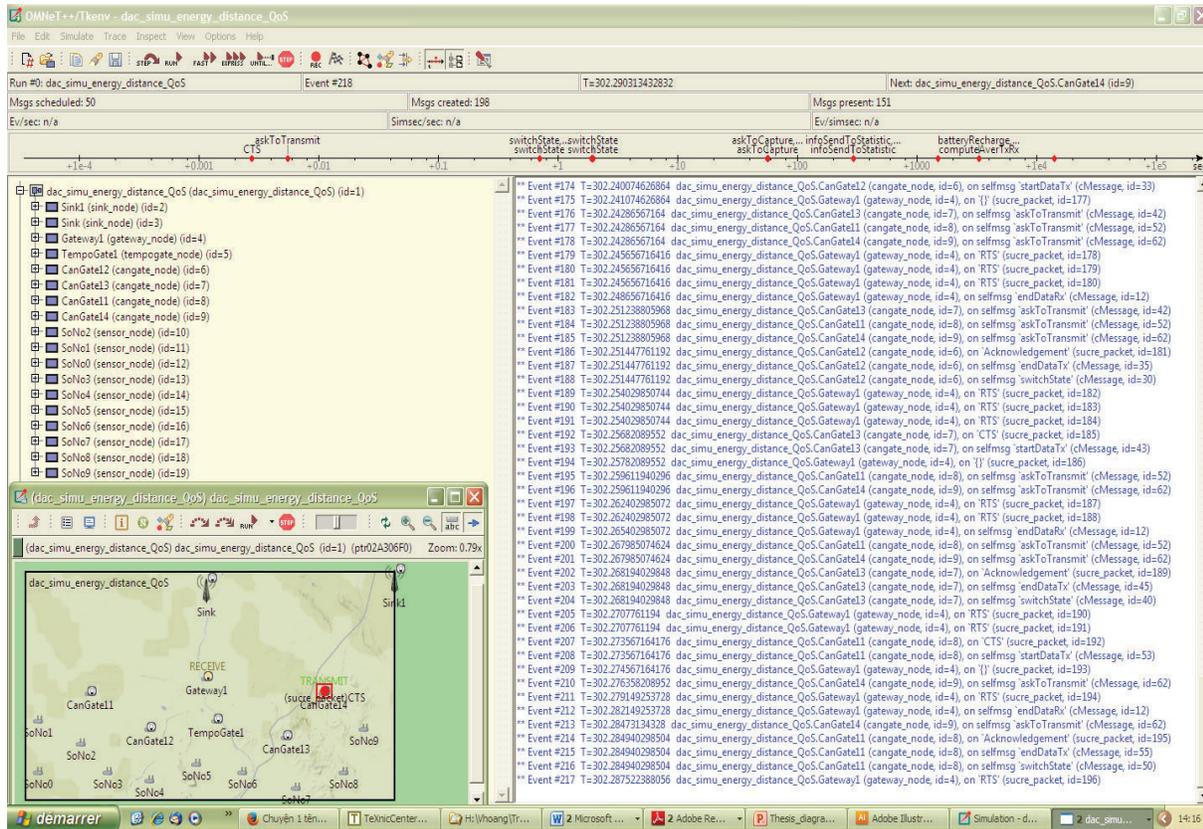


Figure 5.10: Example of a Tkenv user interface in OMNeT++

- Fixed sensing rate of one capture per 60 seconds for all sensor nodes. The Microchip Miwi Pro radio model is used in our simulation, this radio module has $E_{elec} = 890\text{nJ/bit}$ to run the transmitter or receiver circuitry and $\alpha_{amp} = 44\text{pJ/bit}$ for transmission amplifier.
- Gateway nodes are equipped with 1kJ battery, and the remainder nodes are equipped with 500J battery. All sensor nodes are able to scavenge the ambient energies (sun, wind) as described in previous works [18-19].
- Gateway sends a resignation notification to Tempo-CH node when its energy level is less than 1%. And Gateway retakes cluster-head role when its energy level reaches to 10%.
- The cluster-head selection is periodically launched with a time period of one hour (NoOA-CH, P-OACH and PA-OACH).
- The cluster-head selection is launched if number of consecutively failed transmission of cluster head is superior to three (A-OACH and PA-OACH).
- A sequential set of obstacle appearance in the paths between the cluster-head candidate nodes and the Sink is given, in which the position of obstacles changes versus time.

5.5.3 Simulation results

We start the evaluation by looking at the impact of our approach in the network reliability through the reduction of link rupture. Fig.5.11.a shows the reduction of link rupture when applying obstacle-aware protocol strategies. According to this figure, the PA-OACH strategy achieves highest reduction percentage of link rupture (93%) compared to the NoOA-CH strategy. When the reduction percentage of A-OACH (88.6%) is little less than PA-OACH, because the cluster-head selection is only launched when the allowable threshold of consecutively failed transmission of cluster head is violated. The P-OACH strategy has improvement of 59.2% in link rupture reduction due to its ignorance of considering the obstacle appearance during cluster-head period.

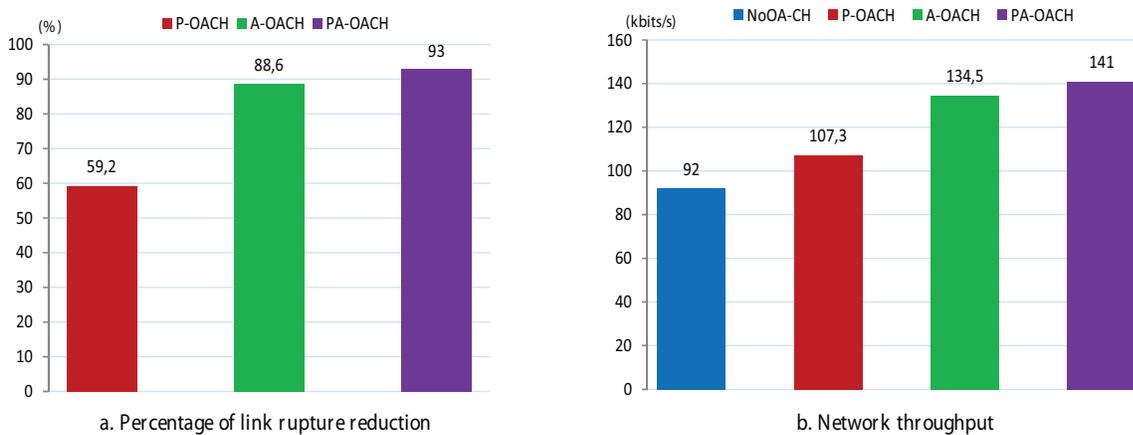


Figure 5.11: Simulation results of network reliability

According to the obtained reduction of link rupture, PA-OACH achieves highest network throughput (141kbits/s), then A-OACH (134.5kbits/s), P-OACH (107.3kbits/s), NoOA-CH (92kbit/s), respectively (as depicted in Fig.5.11.b). Therefore, the PA-OACH strategy improves the network throughput up to 53% compared to the NoOA strategy. Additionally, the throughput of the PA-OACH strategy is equal to 96.5% the network throughput in case of Gateway operation as cluster-head (146kbits/s), which leads to maintain the network reliability.

Next, the network lifetime is also evaluated. There are different definitions for network lifetime, in some applications network lifetime is considered to be the time at which the first node dies, while others consider lifetime to be the time at which last node dies. In our study, we express the network lifetime in term of residual energy in all cluster-head candidate nodes of any cluster. The network lifetime is considered to be the time when one of its clusters is considered as nearly out-of-control, i.e. the remaining battery of all cluster-head candidates of this cluster is less than 1%. The Fig.5.12.a shows that the PA-OACH strategy has the highest improvement of the cluster lifetime, (up to 11% compared to NoOA-CH), due to its highest reduction of link rupture. Thus, the network does not waste much energy for data re-transmission.

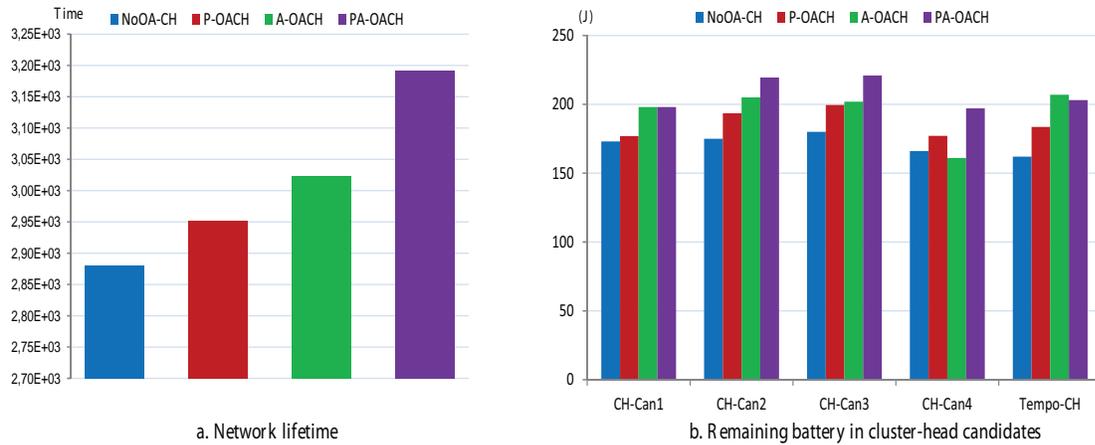


Figure 5.12: Simulation results of network lifetime

To show energy and signaling overheads due to cluster-head selection and the energy distribution in all the cluster-head candidate nodes, wireless network simulation during three months is depicted in Fig.5.12.b. In this figure, the residual energy (including total energy overhead due to cluster-head selection) for five cluster-head candidate nodes of one cluster is plotted after three months of network operation. The PA-OACH strategy balances well energy load between the candidate nodes. Besides, the residual energy of candidate nodes in this strategy is much better than other cases. The average energy overhead and the signaling overhead of cluster-head selection of one cluster for four simulation cases are given in Tab5.2.

Table 5.2: Average energy overhead cluster-head selection of one cluster

	NoOA-CH	P-OACH	A-OACH	PA-OACH
Energy overhead (J)	5.9	7.7	3.8	9.6
Signaling overhead (ms)	39.2	67.2	67.2	67.2

5.6 Conclusion

In this chapter, the issue of cluster-head unavailability when applying the decentralized topology to deploy the network in the harsh environment is presented. Through the existing works, an approach called obstacle-aware cluster head selection is proposed to improve the network lifetime and reliability. Simulations show significant improvements in reduction of link rupture in the network up to 93%, which mitigates the dissipated energy due to re-transmission, consequently, leads to 53% of network throughput improvement and 11% of network lifetime extension. It should be noted that we assume to deal with one failure in cluster at a time in our study:

- If Tempo-CH failure occurs and Gateway is still active, the network continues operating as normal.

- If Gateway is down due to energy depletion or hardware failure, Tempo-CH will replace Gateway and launch the selection of new cluster-head.

In case Tempo-CH is down and Gateway energy is nearly depleted, Gateway will select automatically a candidate node to replace Tempo-CH for launching cluster-head selection.

Chapter 6

DISCUSSION AND CONCLUSION

This work firstly explored energy and availability issues for wireless sensor node and network. Through a detail description of implicit problems related to energy and availability, a novel design was then proposed to manage efficiently the overall energy consumption of battery powered embedded node system, and to monitor operation of primary node components in real-time in order to promptly detect and react against any hardware failure inside sensor node. Next to the simulation experimentation that provides significant improvement of our design in energy-efficiency and availability for wireless sensor node, we realized platforms to analyze power consumption, transition delays and feasibility of our approach. Since the wireless sensor network was deployed in the environment by using cluster topology in our study, we proposed a hierarchical architecture for cluster that eases the monitoring of cluster availability. Based on this cluster architecture, our obstacle-aware cluster-head selection approach was applied to distribute evenly energy load among all sensor nodes in each cluster, and to improve cluster availability. The simulation experimentation shows positive impacts in data throughput, and operating lifetime for wireless sensor network. This chapter presents global discussions and conclusions based on the rich results from this work. The chapter plan is organized as follows: first, the actual simulation experimentation and material implementation approach enabled us to identify the crucial factors from hardware and application specifications, which highly determine the efficiency of our solution in energy and availability improvement for wireless sensor node. Second, since our study targets to medium and large-scale network, we discuss the impact of our cluster-head selection algorithm in term of data throughput, and operating lifetime for these kinds of network. Finally, our conclusion including the main contributions of this work and several perspectives is given.

6.1 Impact Factors on Energy and Availability for Wireless Sensor Node

The following section describes the major factors and application conditions that have been identified to influence the effectiveness of energy reduction, and availability improvement ap-

proach. Actually, four important conditions have been identified in our experiments: the characteristics of operating points, switching latencies of state transitions, sensitivity of current measurement device, and component endurance impact on energy reduction and availability. We discuss each of these points separately in the followings subsections.

6.1.1 Operating Points Impact on Energy Reduction

The characteristics of operating points play important role on energy consumption, because the amount of energy reduction of a given node platform is highly dependent upon its support frequencies and their corresponding supply voltage levels. As explained in section 3.2, the principle of our power strategy is to decrease the frequency of processor that actually results in reduction of energy dissipation. Although this fundamental assumption has been proven in many platforms [M.T. Schmitz, 2004 [3]], in fact there are some cases where it is not. Decreasing frequency decreases power at which a processor operates but at the same time results in an increase of execution time. As energy is the product of power by time, therefore energy reduction is dependent on the condition that the proportion of power decrease is greater than the proportion of time reduction, which depends on the characteristics of operating points in terms of frequency values and associated supply voltage levels. In our study, to avoid overheads in term of energy and time when using DVFS method, we only use two frequency levels that correspond to minimum and maximum supply voltage levels. The simulation experimentation of our power management strategy in section 4.3.1 on the sensor node platform for hazardous gas detection actually results in a decrease of energy consumption. However, it does not mean that our power approach always leads to energy reduction when applying in all kinds of sensor node platforms, because their hardware configurations are different. The power consumptions with and without load for a frequency vary from one platform to another due to different voltages and leakage power. Thus, choosing appropriate frequency levels and corresponding supply voltage levels for a given node platform is very crucial to gain energy savings. Other reasons like poor voltage regulators efficiency, too important switching delay latencies, same level of voltage for different frequencies or performance limitations due an early evaluation platform should be considered. For these reasons, the actual efficiency of operating points in term of frequency/supply voltage characteristics must be addressed before applying any power management strategy.

6.1.2 Switching Latencies of State Transitions

As described in section 4.2, our power management strategy is based on the assumption that voltage/frequency values can be changed instantaneously in function of the type of arrived events. However in reality, it takes time to change the CPU frequency/voltage due to factors such as the internal Phase-Lock-Loop (PLL) locking time, time delay for voltage switching in regulator, and capacitances that exist in the voltage path. The node components such as the processor core, memory, and radio module are unavailable during the frequency transition. A real-time application may be sensitive to this period of unavailability, especially if the frequency is switched at a high rate. Several experimentation with DVFS strategy realized by [K.J. Jabran,

2013 [65]] have shown that decreasing the execution time of tasks below the order of a millisecond alters the effectiveness of the strategy very quickly. In other words, applying DVFS method becomes inefficient when the execution time of tasks becomes close to the delays for changing a processor frequency. For this reason, the latency of frequency switching of a platform should also be considered before implementing any DVFS based strategy. Frequency switching of more than several milliseconds may not always be used in case of critical time applications like real-time systems and video decoding (typically 40 ms frame processing), for example. These latency constraints may prevent the power management effectiveness, and its effect is amplified with the number of frequency switching. Through implementation experimentation in section 4.4.3, our material solution results in $6.2\mu\text{s}$ for frequency/voltage switching that is suitable to be applied in WSN application.

6.1.3 Measurement Sensitivity Impact on Hardware Failure Detection

Our hardware failure detection and localization method was introduced in section 4.2.2. The principle of our approach is to add an additional device for real-time measurement of the node supply current. This device cooperates with the Power and Availability Management (PAM) device in our node architecture, in order to detect the failing component, and then take an appropriate solution to make node system less vulnerable. As presented in hardware implementation, the HALL probe is selected to realized real-time measurement due to its small size for space saving, low-power consumption, accurate measurement with wide current measurement range, high immunity to external interference and low-cost. The realization of hardware failure detection platform indicates that when combining the HALL device with the PAM and digital potentiometer devices, our material solution is able to detect a minimum change of 100mA of measured current. That current gap is efficient to detect any hardware failure inside sensor node for WSN applications in our study such as hazardous gas detection, anti-intrusion. However, when applying this method for ultra-low power WSN applications, several hardware failures may not be detected due to its low current consumption. Thus three solutions are proposed that may be combined together to avoid this issue:

- Additional amplifier circuit can be used to change the output voltage magnitude of HALL device to better match the range of the following reference voltage.
- Additional voltage divider circuit can be used to decrease the reference voltage generated by digital potentiometer to better match the output voltage of HALL device.
- Watchdog timers can complementarily be added to cooperate with our approach, in order to monitor the operation of node components. If one component is inactive during a given time interval, the functioning test is executed for this component.

Another shortcoming of our approach is the high energy overhead related to energy consumption of HALL device. In fact, we have carefully considered this shortcoming by proposing measurement circuit with Shunt resistor or current-sense technique [74] combined with amplifier circuit. However, the main restriction of these solutions is measurement accuracy that is

very difficult to satisfy due to the tolerances of all the variables. The tolerance of resistor can be 1%, 5%, or even higher. Capacitance has its initial tolerance with a degradation on the temperature. The inductance has initial tolerance as well as dependency on the DC biasing current, which makes the inductance vary over a large range. Therefore, the energy/accuracy tradeoff is a difficult question to solve, and depending on purpose of application conditions, designers select an appropriate solution to be applied. In our study, we aim to provide a solution that possesses high measurement accuracy, because accurate hardware failure detection is a critical issue. We assume that our implementation solution of hardware failure detection is just a prototype, and it could be optimized in the industrial process to reduce energy consumption.

6.1.4 Component Endurance Impact on Node System

Another issue is raised concerning the component endurance that may impact on energy reduction and availability of wireless sensor node. Though the PAM, digital potentiometer and current measurement devices are chosen to be highly persistent against failure in our study, this issue is carefully taken into account in our study. The Tab.6.1 depicts the consequences that occur when any complementary component of our approach are down.

Table 6.1: Component endurance impacts on node system

Components	Application	Consequence
PAM	Adjustable regulator	*Output supply voltage is at selected maximum value (3.6V).
	Hardware failure detection	*Hardware failure detection mechanism stops operation.
Digital potentiometer	Adjustable regulator	*Output supply voltage is at selected maximum value (3.6V).
	Hardware failure detection	*PAM is able to detect the failure of digital potentiometer, and deactivates the hardware failure mechanism.
HALL probe	Hardware failure detection	*PAM is able to detect the failure of digital potentiometer, and deactivates the hardware failure mechanism.

As shown in the table above, the sensor node system still operates when any hardware failure occurring to complementary component of our approach. The only consequence is that our sensor node will run in degraded mode in this case.

6.2 Impact Factors on Energy and Availability for Wireless Sensor Network

As explained in Section 2.1, thanks to its advantage for reducing radio communication energy and transmission conflict, cluster topology is used to deployed the network in the environment in our study. Using this topology, the sensor nodes in same cluster communicate with the base station (Sink) through their own cluster-head called Gateway. However, major issue is Gateway unavailability due to energy depletion or hardware failure when using cluster topology, which is extremely critical issue because we will lose the data from all the nodes of a cluster. Therefore, an improved cluster-head selection approach is proposed as explained in section 5.3, in order to extend network lifetime and reliability.

The obtained results of network simulation (section 5.5.3) using OMNET++ tool demonstrated that our approach significantly improved the reduction of link rupture in the network up to 93%, which mitigates the dissipated energy due to re-transmission, consequently, leads to 53% of network throughput improvement and 11% of network lifetime extension. However the network used in this simulation only contains one cluster or seventeen nodes. Since our study targets to medium and large-scale network, this section discuss about the impact of our cluster-head selection algorithm in term of data throughput, and operating lifetime for wireless sensor network when its size is bigger.

To demonstrate the impact of our obstacle-aware cluster-head selection in large-scale network two simulation cases are realized:

- *Non Obstacle-Aware Cluster-Head (No-OACH)*: selects future cluster-head based on residual energy, and distance from cluster-head candidates to the Sink without considering the appearance of obstacles in their path. The cluster-head selection is launched periodically with given time period.
- *Periodic Aperiodic Obstacle-Aware Cluster-Head (PA-OACH)*: is our approach by adding the obstacle-aware criteria in cluster-head selection algorithm. The cluster-head selection is launched periodically with given time period or when actual cluster-head detects number of consecutively failed transmission superior than a given threshold.

As described in Tab.5.1, three strategies are proposed for applying our obstacle-aware cluster-head selection. In this section, we only test our approach with PA-OACH strategy because the environment of network simulation is assumed to contain numerous obstacles. For example, the obstacles are assumed to be the container or container trucks in the seaport in case of hazardous gas detection application. As demonstrated in network simulation results in section 5.5.3, the PA-OACH strategy achieved highest improvement in both reduction of link rupture, data throughput and network lifetime. Therefore, we only test this strategy in this section. The network architecture for simulation is specified and configured in OMNeT++ tool as follows:

- The network includes from two to five clusters, and each cluster consists of 16 sensor nodes, including one Gateway, one Tempo-CH, four cluster-head candidates (CH-Can1, CH-Can2, CH-Can3, CH-Can4), and ten normal nodes (SoNo0...SoNo9).
- Fixed sensing rate of one capture per 60 seconds for all sensor nodes. The Microchip Miwi Pro radio model is used in our simulation, this radio module has $E_{elec} = 890\text{nJ/bit}$ to run the transmitter or receiver circuitry and $\alpha_{amp} = 44\text{pJ/bit}$ for transmission amplifier.
- Gateway nodes are equipped with 1kJ battery, and the remainder nodes are equipped with 500J battery. All sensor nodes are able to scavenge the ambient energies (sun, wind) as described in previous works [18-19].
- Gateway sends a resignation notification to Tempo-CH node when its energy level is less than 1%. And Gateway retakes cluster-head role when its energy level reaches to 10%.
- The cluster-head selection is periodically launched with a time period of one hour (PA-OACH).
- The cluster-head selection is launched if number of consecutively failed transmission of cluster head is superior to three (PA-OACH).
- A sequential set of obstacle appearance in the paths between the cluster-head candidate nodes and the Sink is given, in which the position of obstacles changes versus time.

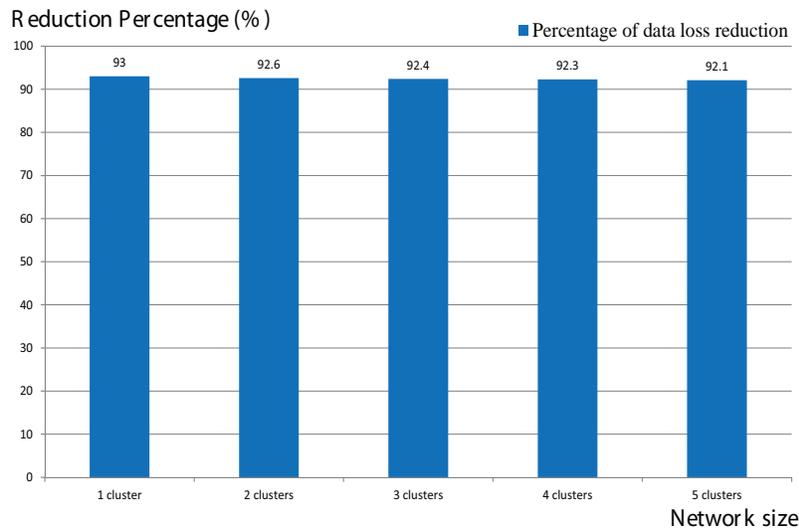


Figure 6.1: Reduction percentage of data loss by applying PA-OACH approach

We start the evaluation by looking at the impact of our approach in the network reliability through the reduction of link rupture. Fig.6.1 shows the reduction of link rupture when applying obstacle-aware protocol strategies. According to this figure, the PA-OACH strategy achieves a

reduction percentage of link rupture of 92%-93% compared to the No-OACH strategy when the network includes from one to five clusters. That demonstrates the stability of our approach when applying for medium or large-scale networks. Another point should be noted that the distribution rate of obstacles is high in our simulation case to evaluate our approach, hence we accept a number of data loss by fixing number of consecutively failed data transmission of cluster head to three for the PA-OACH strategy. That allows avoiding high overhead in term of energy and time delay due to large number of cluster-head selection launching. However, this simulation condition is not always similar to the real environment context where the network is deployed.

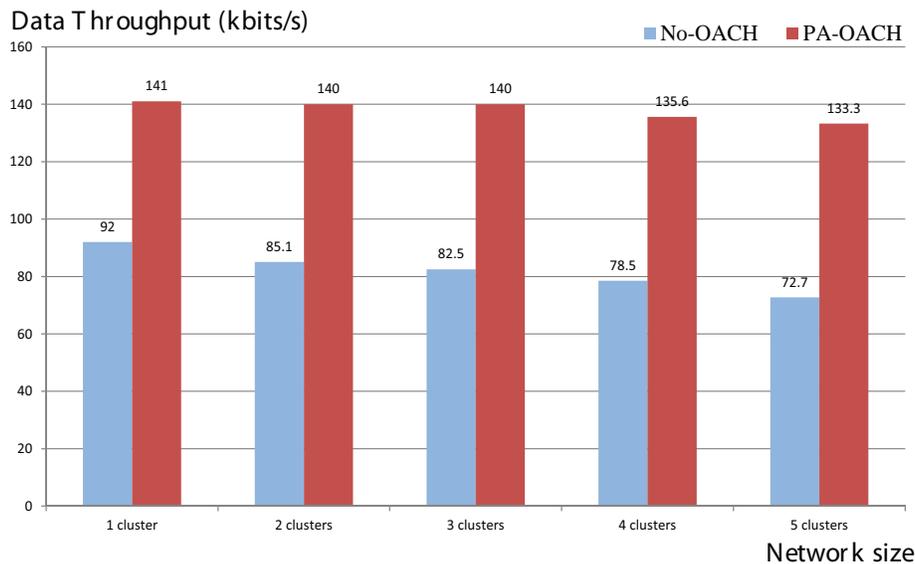


Figure 6.2: Data throughput improvement by applying PA-OACH approach

Next, the data throughput of the network is simulated as seen in Fig.6.2. According to the obtained reduction of link rupture, PA-OACH still maintains a high network throughput (from 133.3kbits/s to 141kbits/s) compared to network throughput of 146kbits/s in case of Gateway operation as cluster-head. Meanwhile, the data throughput is degraded from 92kbits/s to 72.7kbits/s by using the No-OACH strategy when the network size increases from one cluster to five clusters. Therefore, our approach allows maintaining a good service quality for the users.

The network lifetime is also evaluated as depicted in Fig.6.3. To recall, there are different definitions for network lifetime, in some applications network lifetime is considered to be the time at which the first node dies, while others consider lifetime to be the time at which last node dies. In our study, we express the network lifetime in term of residual energy in all cluster-head candidate nodes of any cluster. The network lifetime is considered to be the time when one of its clusters is considered as nearly out-of-control, i.e. the remaining battery of all cluster-head candidates of this cluster is less than 1%.

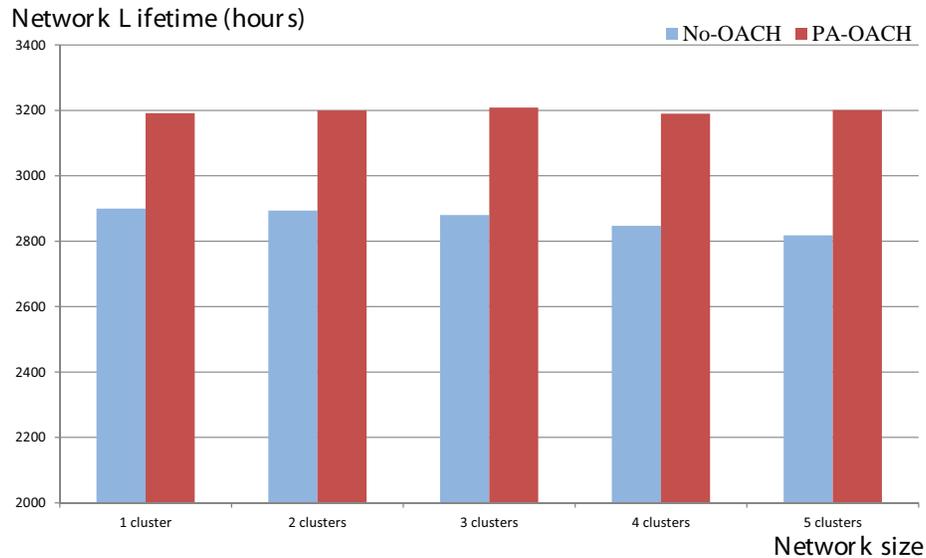


Figure 6.3: Network lifetime improvement by applying PA-OACH approach

The Fig.6.3 shows that the PA-OACH strategy significantly improved the network lifetime compared to the No-OACH strategy. For example, the network lifetime is increase to 11% (1-cluster network), 11.5% (2-cluster, 3-cluster, 4-cluster networks), and up to 13.5% when the network includes five clusters, due to the highest reduction of link rupture of PA-OACH strategy. Thus, the network does not waste much energy for data re-transmission. Briefly, our approach allows improving the lifetime for large-scale network. It should be noted that our sensor node of gas detection application is equipped with 500J battery instead of 84kJ battery in industrial products of ERYMA Company, thus network lifetime in our simulation is about 4.5 months. Additionally, the academic version of OMNET++ tool does not allow network simulation for a long time interval.

The energy distribution in all the cluster-head candidate nodes of each cluster is also considered through a network simulation during three months. The Fig.6.4 and Fig.6.5 present the remaining energy (including total energy overhead due to cluster-head selection) in each candidate nodes of clusters when changing the network size. As seen in these figures, the PA-OACH strategy allows remaining battery energy in each candidate node much higher than those when using the No-OACH strategy. That enables to extend the operating time of the candidate nodes. The average energy overhead and the signaling overhead of cluster-head selection process are given in Tab5.2. The next section provides our conclusions and perspectives for the future works.

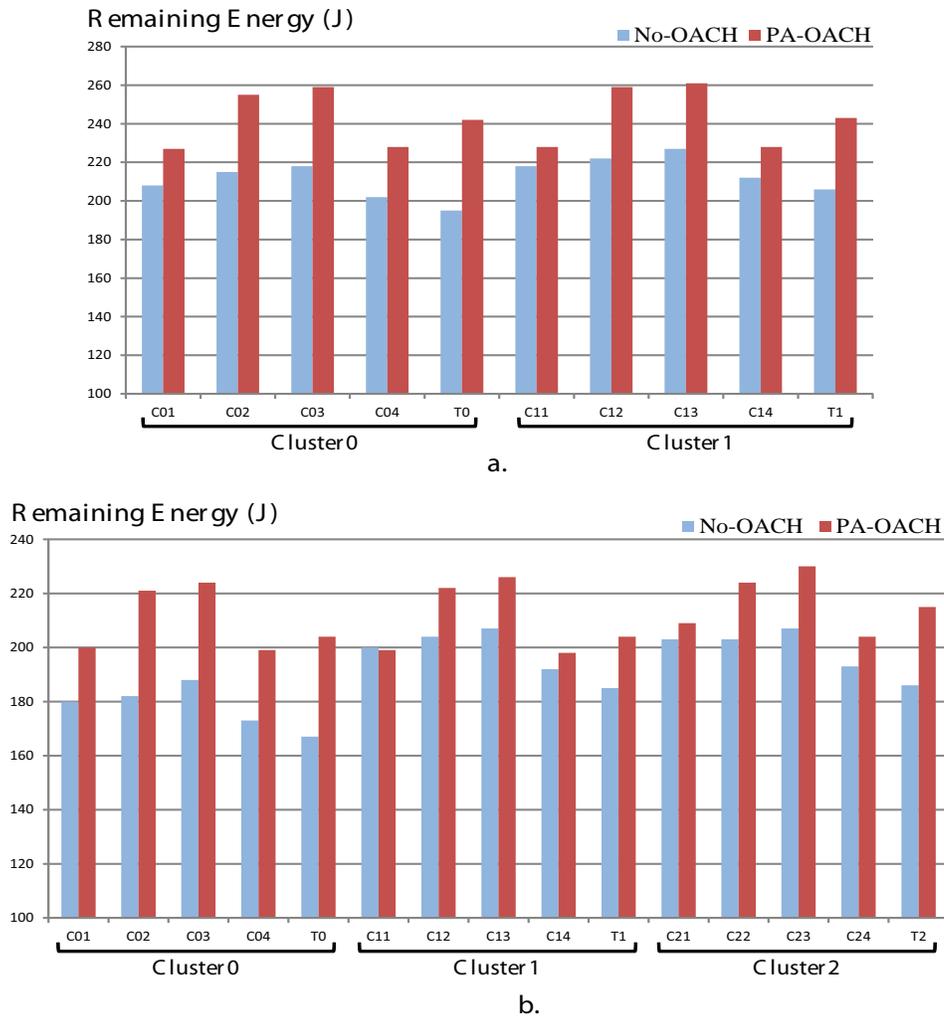


Figure 6.4: Remaining energy of candidate nodes of 2-cluster(a.), 3-cluster(b.) networks

6.3 Conclusion and Perspectives

6.3.1 Conclusion

A great part of our everyday life is somehow linked with cheap and low-power wireless sensor nodes. Since wireless sensor nodes are battery powered devices, the designs of these systems are under increasing pressure to extend battery time. Energy-efficient embedded system, especially low power wireless node system, has become an important challenge for engineering design processes. Many advances in technology enable upcoming generation of embedded system to have lower active and sleep power consumption, while simultaneously increasing the ease of power management development needed to meet time-to-market requirements. Besides, when being deployed in hostile environments, wireless node system faces to implicit hardware failures that highly lead to node crashing, and make sensor node stop operating. These faults are big

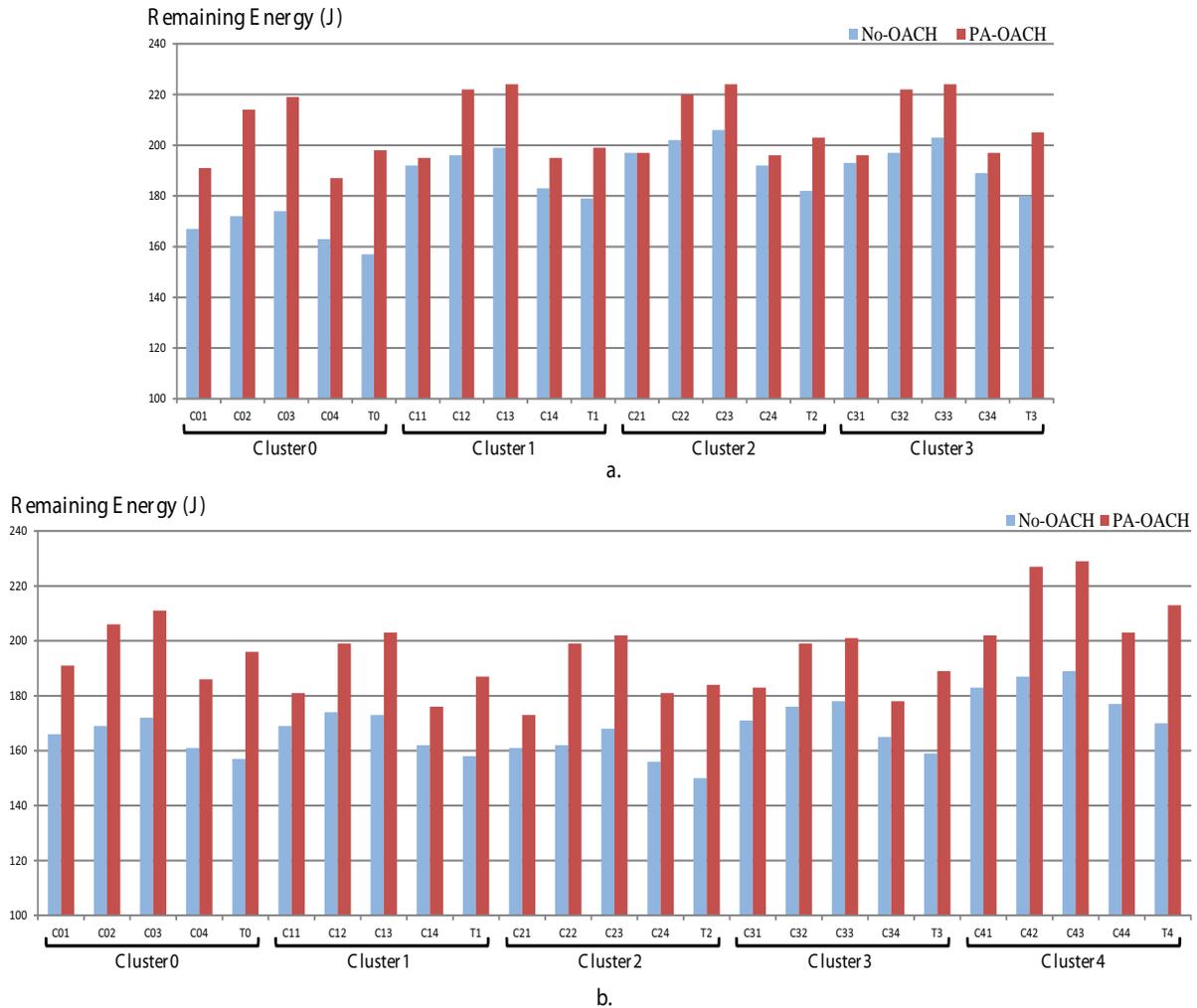


Figure 6.5: Remaining energy of candidate nodes of 4-cluster(a.), 5-cluster(b.) networks

challenge in scientific research in order to provide an appropriate fault detection and diagnosis technique for WSNs.

This thesis has brought a contribution to the characterization of two main constraints in advance power and availability management solutions on wireless sensor node and network. According to our knowledge, this is the first work that targets to node and network layers together for energy-efficiency and availability improvement. Concerning the energy aspect at sensor node level, our state of art pointed out that majority of existing works aim to reduce energy at node layers like communication, network, transport, but a little work focus on physical layer, especially on effective use of DPM and DVFS together. About the availability aspect, we focus on node-self detection approach by providing accurate hardware failure detection based on current measurement, and then it combines with functioning tests to localize failed component. This availability management solution has not existed ever before. Afterwards, we have

further investigated an effective solution in term of energy and performance to improve the network availability during long-term operation. The results obtained in simulation and implementation indicate significant energy-savings accompanied with availability enhancement for both wireless sensor node and network, which demonstrate the effectiveness of our approaches. The most relevant achievements in addressing the effectiveness of power and availability management are summarized below.

- Providing novel design including a Power and Availability Management (PAM) combining with additional components to manage the effective use of energy, and to improve availability for wireless sensor node.
- Realizing test benches to implement the PAM in sensor node by using microcontroller or Actel FPGA Nano family. The obtained results showed that Actel FPGA IGLOO Nano family is much higher energy-efficient than microcontroller when realizing the same task. However, the more complex task PAM has to realize, the more energy it dissipates in active state when implementing it in IGLOO Nano chip due to additionally required logic sources, i.e. its advantages in term of energy-savings is smaller. Related to processing time to response to any arrived event, microcontroller reacts faster than IGLOO Nano controller.
- Reducing power consumption of wireless sensor node by using DPM and DVFS techniques together, which are controlled by PAM. That leads to extend node battery life, i.e. the operating lifetime of wireless node. However, in practice the efficiency of DVFS policy is highly upon the characteristics of operating points of the target platform. These characteristics should be carefully considered before applying DVFS-based strategy.
- Realizing the test bench of adjustable regulator for applying DPM and DVFS techniques. This low-power regulator allows tuning dynamically the supply voltage during node operation with a short time delay.
- Improving node availability by using PAM to monitor the operation of primary node components, in order to detect any hardware failure inside node system, and to take appropriate solution to keep sensor node still operating.
- Realizing test bench for hardware failure detection method, its principle is based on on-line current measurement using HALL probe device. This small, low-power and low-cost device allows node system to detect accurately any hardware failure issue with minimum change of 100mA of measured current. To apply this approach in ultra-low power wireless sensor node, several additional hardwares need to be added to change signal magnitude to better match the range of the following reference voltage, or the watchdog timers can be used to cooperate with our approach, in order to monitor the operation of node components.
- Providing an obstacle-aware cluster head selection algorithm to deal with cluster-head unavailability issue. This algorithm is able not only to consider remaining energy, distance-to-Sink, but also obstacles appearing during network operation, in order to choose an

appropriate sensor node for undertaking cluster-head role. Simulation results showed significant improvement in reduction of link rupture in the network that reduces energy dissipation due to re-transmission. As a result, the network throughput and lifetime is highly increased. Our approach can be applied for small, medium or large-scale wireless sensor network.

6.3.2 Perspectives

This work was carried out in cooperation with ERYMA Company which offers an expertise in material solutions and security system for the prevention, monitoring, maintenance and remote control. We aim to address the challenge of finding a novel design for wireless sensor node to reduce energy consumption and enhance node availability. Next, we focus on energy-efficiency and availability of wireless network by proposing an improved cluster-head selection approach. In the future works, several points of view should be considered.

In the short term at node level:

- Our contribution will focus on how to implement effectively the power and availability management in a real node platform, which will be built in industrial process. Therefore, an axis of research will be to address for several node architectures of different WSN applications based on the use of dynamically reconfigurable embedded system, and verify their operation when deployed in the different monitoring environments.
- In case of the PAM implementation in FPGA chip, concerning the trade-off between performance and energy consumption, what kind of implementation (soft-core or hard-core) is consistent?
- To implement our hardware failure detection and localization in real sensor node system. Another research should be carried out to find appropriate location in node circuit to place HALL probe device to provide accurate measurement.

In the long term at node level:

- Since the PAM device is the most important element of our approach, its abilities of noise immunity, magnetic immunity, and endurance against hardware failure should be carefully considered when deploying sensor node in harsh environments. After long operation time of experimental tests in these environment conditions, we could determine what kind of microcontroller or FPGA chip that is suitable for implementing the PAM.

In the short term at network level:

- To deploy efficiently our obstacle-aware cluster-head selection in the real network, the synchronization of launching process of cluster-head selection with cluster-head candidate nodes is very necessary, in order to guarantee all candidate nodes to participate in cluster-head selection procedure.

- Since the number of candidate nodes to be set in each cluster is dependent on the monitoring environment, hence some case studies must be considered to find the most suitable number of candidate nodes should be selected during network deployment, in order to achieve highest energy-efficiency, reliability and operating time for the clusters.
- Since the efficiency of each strategy (P-OACH, A-OACH, PA-OACH) depends on the environment context, where the network is deployed. Under autonomic vision, each cluster should automatically select the most suitable strategy to apply according to the change of environment context. For example, in the application of hazardous gas detection in the harbor, the rate of obstacle appearance (container trucks) is very high from 7am to 7pm and very low even zero from 7pm to 7am. Thus the cluster can select the PA-OACH strategy in the first period (from 7am to 7pm), and switches to P-OACH or A-OACH in the second period, which leads to reduce the energy overhead due to cluster-head selection.

In the long term at network level:

- The impact of increasing battery capacity (2x, 3x or more) of Gateway node in the network structure and operation should be considered. For example, we could consider two case studies. In the first case, only three nodes in each cluster are selected to be in the set of cluster-head candidates when increasing the Gateway battery capacity to two times (2x). And in the second case, only two nodes in each cluster are selected to be in the set of cluster-head candidates when increasing the Gateway battery capacity to three times (3x). Then we compare the impact of each case to network operation (data throughput, lifetime), in order to determine which case is better.
- As the positions of cluster-head candidate nodes in their cluster may influence the effectiveness of the cluster-head selection, an axis of research related to operational research domain should be studied, in order to find the best positions for cluster-head candidate nodes to be located in their clusters.
- To propose another approach for addressing the Sink unavailability issue without adding redundant Sink node.

6.4 List of Publications

Publications in National Conferences:

1. *Conception sous contraintes de sûreté de fonctionnement et de consommation d'énergie de réseau de capteurs sans fil*, GDR SoC-SiP national conference, June 13-15, 2012, Paris, France.
2. *The detection and localization of hardware failure for Wireless Sensor Node based on online power management*, GDR SoC-SiP national conference, June 11-13, 2013, Lyon, France.

Publications in International Conferences:

1. *Design under Constraints of Availability and Energy for Sensor Node in Wireless Sensor Network*, International Conference on Design and Architectures for Signal and Image Processing (DASIP), October 2012, Karlsruhe, Germany.
2. *On-line self-diagnosis based on power measurement for a wireless sensor node*, First Workshop on Highly-Reliable Power-Efficient Embedded Designs, February 24th, 2013, Shenzhen, China.
3. *Increasing the autonomy of Wireless Sensor Node by effective use of both DPM and DVFS methods*, 12th Edition of IEEE Faible Tension Faible Consommation - IEEE FTFC 2013 in Paris, France.
4. *Cluster-Head Selection Algorithm to Enhance Energy-Efficiency and Reliability of Wireless Sensor Networks*, 20th European Wireless (EW) Conference, May 2014, Barcelona, Spain.

Bibliography

- [1] Peter Harrop and Raghu Das, *Wireless Sensor Networks (WSN) 2012-2022: Forecasts, Technologies, Players - The new market for Ubiquitous Sensor Networks (USN)*, December, 2012.
- [2] M. Weiser, *The computer science issues in ubiquitous computing*, Communication of ACM, July, 1993, Vol.36, No.7, pp. 75–84.
- [3] Marcus T. Schmitz, Bashir M. Al-Hashimi, Petru Eles, *System-Level Design Techniques for Energy-Efficient Embedded Systems*, Kluwer Academic Publishers, first edition, Boston, USA, 2004.
- [4] W. Dargie, C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*, Wiley Series on Wireless Communications and Mobile Computing, 2010.
- [5] Chandrakasan, A.P. and Brodersen, R.W., *Low power digital CMOS design*, IEEE Journal of Solid-State Circuits, April, 1992, pp. 54–64.
- [6] G.E. Moore, *Cramming more components onto integrated circuits*, Reprinted from Electronics, volume 38, number 8, April 19, 1965, Solid-State Circuits Society Newsletter, IEEE, Volume:11, 2006.
- [7] Y.C. HO, *Introduction to special issue on dynamics of discrete event systems*, Proceedings of the IEEE, 1989.
- [8] P.J.G. Ramadge and W.M. Wonham, *The control of discrete event systems*, Proceedings of the IEEE, 1989.
- [9] G. Cassandras, *Discrete event systems, modeling and performance analysis*, Aksen Associates Incorporated Publishers, 1993.
- [10] S. Lafortune, C.G. Cassandras, *Introduction to Discrete Event System*, Springer Publishers, second edition, Harvard, USA, 2008.
- [11] D. Harel, *Communications of the Association Computing Machinery*, On visual formalisms, 1988.

- [12] J.L. Peterson, *Petri Net theory and the modeling of systems*, Englewood cliffs. Prentice Hall, 1981.
- [13] Khaled Arisha, Moustafa Youssef, and Mohammed Younis, *Energy-Aware TDMA-Based MAC for Sensor Networks*, System-Level Power Optimization for Wireless Multimedia Communication, Kluwer Academic Publishers, 2002.
- [14] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, *Energy-efficient, collision-free medium access control for wireless sensor networks*, In SenSys, Los Angeles, California, USA, 2003.
- [15] Tijs van Dam, Koen Langendoen, *An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks*, The First International Conference on Embedded Networked Sensor Systems, New York, NY, USA 2003.
- [16] Mo Sha, Guoliang Xing, Gang Zhou, Shucheng Liu, Xiaorui Wang, *C-MAC: Model-driven Concurrent Medium Access Control for Wireless Sensor Networks*, The Twenty-Eighth Conference on Computer Communications, Rio de Janeiro, Brazil, 2009.
- [17] Hossam Hassanein, Jing Luo, *Reliable Energy Aware Routing in Wireless Sensor Networks*, The Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS), Columbia, Maryland, USA, 2006.
- [18] Lopez-Gomez, M.A., Tejero-Calado, J.C., *A Lightweight and Energy-Efficient Architecture for Wireless Sensor Networks*, IEEE Transactions on Consumer Electronics, 2009.
- [19] C. Srisathapornphat, C.-C. Shen, *Coordinated Power Conservation for ad-Hoc Networks*, Proc. IEEE International Conference on Communications, IEEE ICC 2002, vol. 5, 2002.
- [20] V. Tsaoussidis and H. Badr, *TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains*, Proceeding of International Conference on Network Protocols, Osaka, Japan, 2000.
- [21] J. Liu and S. Singh, *ATCP: TCP for Mobile Ad-Hoc Networks*, IEEE JSAC, Wireless Commun. Series, vol. 19, no. 7, 2001.
- [22] S. Agrawal and S. Singh, *An Experimental Study of TCP Energy Consumption Over a Wireless Link*, The Fourth European personal Mobile Communication Conference, Vienna, Austria, 2001.
- [23] A. Sinha, and A. Chandrakasan, *Dynamic Power Management in Wireless Sensor Networks*, IEEE Design and Test of Computers, vol. 18, Issue 2, 2001.
- [24] G. Ciardo, A. Blakemore, P. Chimento, JR. Jogesh, K. Muppala and K.S. Trivedi, *Automated generation and analysis of markov reward models using stochastic reward nets*, Linear Algebra, Markov Chains and Queuing Models, Eds: Springer Verlag, 1993, pp. 145–191.

- [25] Molloy, M.K., *Discrete Time Stochastic Petri Nets*, IEEE Transactions on Software Engineering, 1985.
- [26] Nicolas Ferry , Sylvain Ducloyer , Nathalie Julien and Dominique Jutel, *Power/Energy Estimator for Designing WSN Nodes with Ambient Energy Harvesting Feature*, EURASIP Journal on Embedded Systems, January , 2011.
- [27] Nicolas Ferry, Sylvain Ducloyer, Nathalie Julien and Dominique Jutel, *Energy Estimator for Weather Forecasts Dynamic Power Management of Wireless Sensor Networks*, Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation, Springer Publishers, 2011.
- [28] J. Muppala, G. Ciardo, K.S. Trivedi, *Spnp: Stochastic petri net package*, Proceedings of the Third International IEEE Workshop, 1989, pp. 142–151.
- [29] J. Muppala, G. Ciardo, K.S. Trivedi, *Stochastic Reward Nets for Reliability Prediction*, Communications in Reliability, Maintainability and Serviceability, 1994.
- [30] B.S. Deepaksubramanyan and Adrian Nunez, *Analysis of Subthreshold Leakage Reduction in CMOS Digital Circuits*, PROCEEDINGS OF THE 13TH NASA VLSI SYMPOSIUM, POST FALLS, IDAHO, USA, June 5-6, 2007.
- [31] Bhatti, M.K., Belleudy, C. and Auguin, M., *An inter-task real time DVFS scheme for multiprocessor embedded systems*, IEEE Conference on Design and Architectures for Signal and Image Processing (DASIP), Edinburgh, England, October 26-28, 2010.
- [32] S. Harte, A. Rahman and K. Razeeb, *Fault tolerance in sensor networks using self-diagnosing sensor nodes*, in The IEEE International Workshop on Intelligent Environments, June 2005, pp. 7–12.
- [33] A. Mahapatro and P.M. Khilar *Detection of node failure in wireless image sensor networks*, ISRN Sensor Networks, 2012.
- [34] A. Mahapatro and P.M. Khilar *Energy-efficient distributed approach for clustering-based fault detection and diagnosis in image sensor networks*, IET Wireless Sensor Systems, 2013.
- [35] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, *A discrete-time battery model for high-level power estimation*, Proceedings of the conference on Design, automation and test in Europe (DATE'00), 2000.
- [36] K. Casey, A. Lim and G. Dozier, *A Sensor Network Architecture for Tsunami Detection and Response*, International Journal of Distributed Sensor Networks, vol. 4, January 2008, pp. 28–43.
- [37] M. Yu, H. Mokhtar and M. Merabti, *Fault management in wireless sensor networks*, IEEE Wireless Communications, vol. 14, December 2007, pp. 13–19.

- [38] F. Koushanfar, M. Potkonjak and A. Sangiovanni-Vincentelli, *Fault tolerance techniques for wireless ad hoc sensor networks*, Proceedings of IEEE Sensors, vol. 2, 2002.
- [39] L. M. S. D. Souza, H. Vogt and M. Beigl, *A survey on fault tolerance in wireless sensor networks*, 2007.
- [40] K. Martinez, P. Padhy, A. Riddoch, H. Ong, and J. Hart, *Glacial environment monitoring using sensor networks*, In REALWSN, 2005.
- [41] A. Mahapatro and P.M. Khilar, *Fault diagnosis in wireless sensor networks: a survey*, IEEE Communications Surveys and Tutorials, vol. 15, Mars 2013.
- [42] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, *Lessons from a sensor network expedition*, in Wireless Sensor Networks, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2004.
- [43] Wei Ye, J. Heidemann, D. Estrin, *An energy-efficient MAC protocol for wireless sensor networks*, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.
- [44] Wendi R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, USA, 2000.
- [45] Liu YH, Gao JJ, Jia YC, Zhu LG, *A cluster maintenance algorithm based on LEACH-DCHS protocol*, Proc. international conference on networking, Chongqing, China, June 2008.
- [46] Solaiman Ali Md, Dey Tanay, Biswas Rahul, *ALEACH advanced LEACH routing protocol for wireless microsensor networks*, International Conference on Electrical and Computer Engineering, ICECE, 2008.
- [47] Y. Liang, H. Yu, *Energy Adaptive Cluster-Head Selection for Wireless Sensor Networks*, Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2005.
- [48] Y. Liu, Y. Zhao, *A New Clustering Mechanism Based On LEACH Protocol*, International Joint Conference on Artificial Intelligence, 2009.
- [49] K. Ramesh, Dr. K.Somasundaram, *A comparative study of clusterhead selection algorithms in wireless sensor networks*, International Journal of Computer Science and Engineering Survey (IJCSSES) Vol.2, No.4, November 2011.
- [50] Salhieh A., Weinmann J., Kochhal M., Schwiebert L., *Power efficient topologies for wireless sensor networks*, IEEE International Conference on Parallel Processing, Valencia, Spain, September 2001.

- [51] Wei Ye, J. Heidemann, D. Estrin, *An energy-efficient MAC protocol for wireless sensor networks*, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.
- [52] Chee-Yee Chong and Kumar, S.P., *Sensor networks: evolution, opportunities, and challenges*, Proceedings of the IEEE, August 2003.
- [53] David C. Steere, Antonio Baptista, Dylan McNamee and Calton Pu Jonathan Walpole, *Research challenges in environmental observation and forecasting systems*, Proceedings of the 6th annual international conference on Mobile computing and networking MobiCom '00, 2000.
- [54] M. Ringwald, K. Romer, *Deployment of sensor networks: Problems and passive inspection*, Fifth IEEE Workshop on Intelligent Solutions in Embedded Systems, June, 2007.
- [55] Laurent, J., Senn, E., Julien, N., Martin, E., *Functional Level Power Analysis: An Efficient Approach for Modeling the Power Consumption of Complex Processors*, Proceedings of the conference on Design, automation and test in Europe, DATE, 2004.
- [56] Berruet P., Kindler E., *Nested Anticipation in Design Reconfigurable Manufacturing Systems*, International Journal of Control, Automation, and Systems, 2006.
- [57] Jimy Tatiana RASOLOFONIRINA, Eric SENN, Mickael LANOE, *Etude de la consommation des systemes Embarquees a base de FPGA*, Master 2 thesis, 2012.
- [58] Lucile Senn, Eric SENN, Mickael LANOE, *Etude des performances des processeurs embarqués NIOS sur plateforme FPGA*, Master 2 thesis, 2011.
- [59] Low K. S., Win W.N.N., Meng Joo Er, *Wireless Sensor Networks for Industrial Environments*, International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 2005.
- [60] Serodioa C., Cunhab J. B., Moraisb R., Coutoc C., Monteiro J., *A networked platform for agricultural management systems*, Computers and Electronics in Agriculture, 2001.
- [61] Ribeiro A. et al., *A friendly man-machine visualization agent for remote control of an autonomous tractor GPS guided*, The Proceedings of the 4th European Conference in Precision Agriculture, Berlin, Germany, 2003.
- [62] Damas M., Prados A.M., Gomez F., Olivares G., *HidroBus system: fieldbus for integrated management of extensive areas of irrigated land*, Microprocessors Microsyst., 2001.
- [63] Monton E., Hernandez J.F., Blasco J.M., Herve T., Micallef J., Grech I., Brincat A., Traver V., *Body area network for wireless patient monitoring*, IET Communications, 2008
- [64] Murat Miran Koksall, *A Survey of Network Simulators Supporting Wireless Networks*, October, 2008.

- [65] Jabran KHAN JADOON, *Evaluation of power management strategies on actual multiprocessor platforms*, March, 2013.
- [66] *FPGA IGLOO nano*, Actel Corp., http://www.actel.com/documents/IGLOO_nano_DS.pdf.
- [67] *Libero IDE Quick Start Guide for Software v9.0*, Microsemi Corporation.
- [68] *AD5204 digital potentiometer*, http://www.analog.com/static/imported-files/data_sheets/AD5204_5206.pdf.
- [69] *PTH04070W digital regulator*, <http://www.ti.com/lit/ds/slts227b/slts227b.pdf>.
- [70] *Streetline, Inc.*, <http://www.streetline.com>, 2014.
- [71] <http://ww1.microchip.com/downloads/en/DeviceDoc/51589a.pdf>.
- [72] <http://www.omnetpp.org/doc/omnetpp/manual/usman.html>
- [73] <http://www.farnell.com/datasheets/65365.pdf>.
- [74] <http://dangerousprototypes.com/2014/07/13/app-note-a-simple-current-sense-technique-eliminating-a-sense-resistor/>
- [75] <http://www.eryma.com>.
