Multi-Objective Genetic Algorithms for Computing Fast Fourier Transforms over Wireless Sensor networks

I. Njini

00000002-6220-459x

BSc(Hons) M., PGD O R., MSc CS

A Thesis Submitted in Fulfilment of the Requirements for the award of the Degree Doctor of Philosophy(PhD) in Computer Science

Department of Computer Science

School of Mathematical and Physical Science

Faculty of Agriculture, Science and Technology

North-West University, Mafikeng Campus

Supervisors Professor Obeten O. Ekabua
Professor Michael Esiefarienrhe

Graduation October 2017

Student number: 24045098

http://www.nwu.ac.za/

It all starts here ™

NORTH-WEST UNIVERSITY ®
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

Declaration

I declare that this research study on **Multi-Objective Genetic Algorithms for Computing Fast Fourier Transforms over Wireless Sensor Networks** is my work and has not been presented for the award of a degree in this, or any other university. All the information used has been duly acknowledged both in text and in the final references.

Signature____________________ Date 19/10/2017

Approval:

Signature____________________ Date 16/10/2017

Supervisor: **Professor O.O. Ekabua**

Department of Computer Science

Faculty of Agriculture, Science and Technology

North-West University, Mafikeng Campus

South Africa

Signature____________________ Date 19/10/2017

Supervisor: **Professor M. Esiefarienhe**

Department of Computer Science

Faculty of Agriculture Science and Technology

North-West University, Mafikeng Campus

South Africa

Dedication

This Thesis is dedicated to my wife and children for the love and support they gave me throughout my research and studies.

Acknowledgements

I take this opportunity to thank God for his protection and guidance throughout the time of my research studies.

I acknowledge and express my sincere gratitude to my supervisor, Professor O.O. Ekabua for his intellectual guidance and inspirational support and advice, without which this research could not have been a success. Professor O.O. Ekabua's passion for research has inspired me from the time I enrolled for research studies at North-West University and it shall continue to do so throughout my academic and professional work. I also appreciate the detailed inspection and contribution of my core supervisor, Professor Michael B. Esiefarienhe.

I thank and acknowledge all staff members of the North-West University, especially the Department of Computer Science, for the support that they rendered me throughout the period of my studies.

I acknowledge all my PhD colleagues for the encouragement and emotional support.

Finally, I praise the Lord, the Almighty for giving me this opportunity to study at this level.

Abstract

Wireless sensor networks (WSNs) consist of sensor nodes which are geographically distributed with the capacity to communicate and re-organize themselves within the network. Each sensor node possesses wireless communication capabilities in the form of radio signals and some level of intelligence for signal processing and networking data. In most practical implementations, WSNs are deployed with limited battery sources and it is time consuming to replace the battery sources in practice which constrain the energy requirements of the WSN and inhibit network performance. While researches in traditional WSN applications are primarily focused on achieving high quality of service provisions, this current research focuses on protocols aimed at power conservation. This research initially identifies and describes the nature of problems that arise in deploying WSNs due to their limited battery sources and complications that arise in replacing the battery sources in practice. Therefore, we explore problems associated with sensor communication and distributed processing in WSNs. Our initial approach was to review existing techniques from literature applicable to distributed signal processing that resolves the challenge associated with efficient energy consumption in WSNs. The review indicates that data redundancy in communication is significantly due to spatio-temporal correlations. Among other techniques used, comprehensive sensing technique was discovered to be more effective in reducing the amount of redundant data communicated. Also, the literature survey indicates that the bulk of WSNs energy consumption is principally due to ineffective node communication resulting in poor load balancing. Furthermore, the bulk of energy expended during that ineffective node communication is associated with complex number multiplication. Yet, existing literature has failed to handle these issues of ineffective node communication resulting in poor load balancing and complex number multiplication which leads to increased energy depletion. Consequently, our novel technique is through a simulated approach where we developed a solution through the application of Multi-Objective Genetic Algorithm for Field Programmable Gate Arrays in order to resolve the challenge of complex number multiplication during signal processing at the node level. In handling the challenge of ineffective node communication, resulting in poor load balancing between nodes, we design a stochastic context-aware energy consumption and distribution model that effectively resolves the problem of load balancing. In sum, this research work contributes to knowledge in that it has developed a Multi-Objective Genetic Algorithm based on field programmable gate arrays to resolve the challenge of complex multiplication associated with the computation of Fast Fourier Transform during signal processing at the node level. Another novel technique achieved by this research is the design of Stochastic Context-aware energy consumption and distribution model capable of resolving the problem of load balancing in a WSN. These two novel techniques are aimed at reducing computational complexity, hence energy consumption during signal processing in WSNs.

DEFINITION OF TERMS

Genetic Algorithms: Genetic algorithms are search procedures modeled on the Mechanism of Natural selection and evolution. They use the techniques found in nature such as reproduction, gene crossover and mutation to find optimal solutions to mathematical problems.

Discrete Fourier Transforms: Discrete Fourier Transforms (DFT) takes as input discrete signals and converts the signals from the time domain (Signal Strength as a function of time) to the frequency domain (Signal strength as a function of frequency). It shows the signal's spectral content, divided into discrete frequency bands (bins).

Inverse Discrete Fourier Transform: The Inverse Discrete Fourier Transforms are the reverse process to the Discrete Fourier transform (DFT). They enable us to recover the original discrete signal from the frequency samples.

Fast Fourier Transform: The Fast Fourier transforms are algorithms for computing Fourier transforms. They are faster than the DFTs.

Distributed Source Coding: Distributed source coding refers to the compression of multiple correlated output data from several sensors that do not communicate with each other. Each sensor quantizes its input data separately without inter sensor communication and transmits the results to the decoder which reconstructs the observation from the several correlated sensors jointly.

Wireless Sensor Network: This is a collection of communicating micro-electro mechanical devices deployed in some geographic region for the purposes of sensing and transmitting data.

LIST OF ACRONYMS

DSNs:	Distributed Sensor Networks
FPGA:	Field Programmable Gate Array
SNR:	Signal to Noise Ratio
SA:	Switching activity
CS:	Compressive Sensing
RIP:	Restricted Isometry Property
NPH:	Non-Polynomial Hard
BP:	Basis pursuit
BPD:	Basis Pursuit De-noising
LASSO:	Least Shrinkage and Selection Operator
LARS:	Least Angle Regression
COSAMS:	Compressive Matching Sampling Pursuit
VHDL;	Verilog Hardware Decision Language
MC-CDMA:	Multi-carrier code Division Multiple access
OFDMA:	Orthogonal Frequent Division Multiple Access
SCAEDM:	Stochastic Content Aware Energy Consumption Distance Model
CAD:	Computer Aided Design
SOC	System on Chip
LUT:	Programmable Logic Unit
PE:	Processing Elements
R4SDC:	Radix- 4 Single part Delay Commutator
CORDIC:	Coordinate Rotation Digital Commutator
MOGAFPGA:	Multi-Objective Genetic Algorithm Based Field Programmable Gate Array
EAs:	Evolutionary Algorithms
OR:	Opportunistic Routing.

PRR	Packet Reception Ratio
LQM	Link Quality Metric
SDFD	Stochastic Function for Energy Delay Distribution
RSSI	Received Signal Strength Indicator
CDS	Connected Dominating Set
LBCDC	Load balanced Connected Dominating Set
$SDFD_{max}$	Predefined maximum delay at each node.
ECDF	Energy Consumption Distribution Function.
RE	Residual Energy
CMOS	Complementary metal-oxide-semiconductor

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

1.0 Background Information

Consider an ad hoc wireless sensor network with several distributed micro electro mechanical devices (sensor nodes) deployed in some geographical region and they can reorganize themselves within the network. Each sensor node has wireless communication capabilities, usually in the form of radio signals and some level of intelligence for signal processing and networking data [1]. Some modern smart sensors comprise of a sensing unit, a processing unit, a communication unit, a location finding unit, a mobiliser and a power generating unit. Some of the components such as location finding unit, Mobiliser unit and power generation unit are optional and application dependent. Such component may not be available in some common sensor applications. Figure 1.1 shows the main components of a smart sensor node.

In most practical implementations, WSNs are normally deployed with limited and practically irreplaceable battery sources. This in turn puts a limiting constraint on the energy requirements of the WSN application, thereby inhibiting the operations network. While traditional network applications focused primarily on achieving high quality of service provision, current research focuses on protocols aimed at power conservation. They must have in-built trade-off mechanisms that give the end user the option of prolonging network lifetime at the cost of low power throughput or high transmission delay [2].

Comprehensive evaluation of current state of the art sensor motes performance is presented in [3]. Sensor motes currently available in the market were evaluated in terms of prices, software portability and performance evaluation. The researcher reported that software portability remains one of the main design challenges as most of the currently available hardware design were developed to meet certain predefined functional requirements. Figure 1.2 and 1.3 presents some of the commonly available sensor motes.

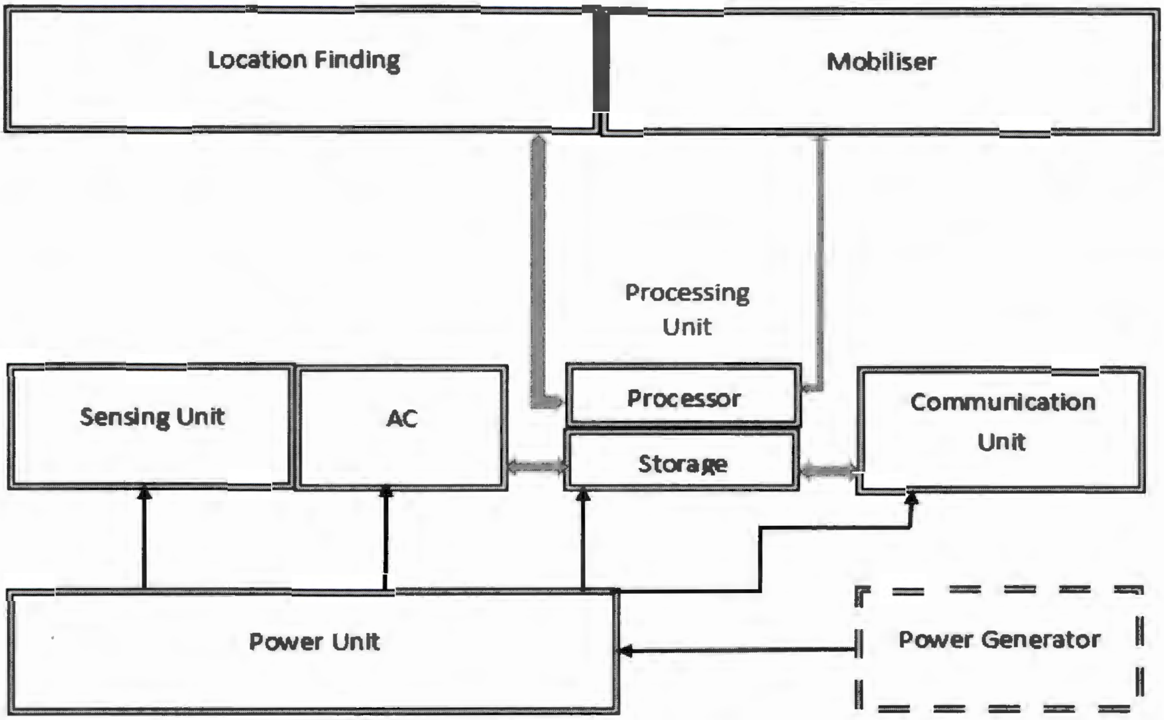


Figure 1.1: Components of Smart Sensors Nodes

Traditional network designs have been influenced mainly by factors such as operating environment, fault tolerance, scalability, hardware constraints, transmission media, network topology, and power consumption. These factors have been addressed by many researchers but, to my knowledge, most of the researches have concentrated on energy conservation in routing protocols. In contrast, this research looked at achieving energy conservation through sensor collaboration and distributed processing. While less information is lost when communication is at a lower level (e.g. raw signals) being sent from source to destination node (sink), this usually results in more bandwidth requirements and a lot of data redundancy, as duplicate information is sent.

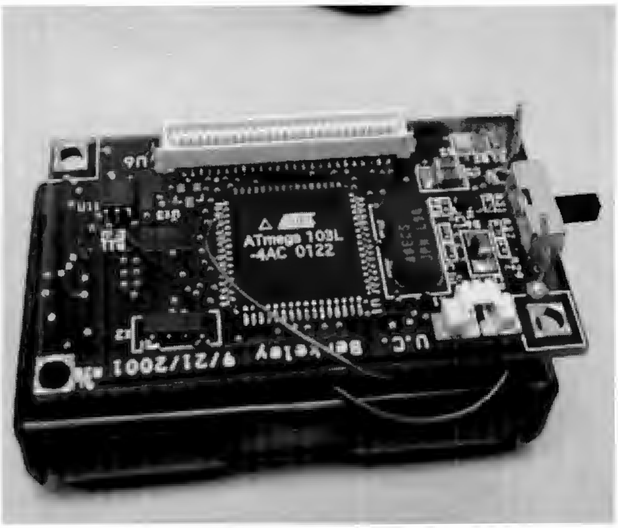


Figure 1.2: Mica Motes

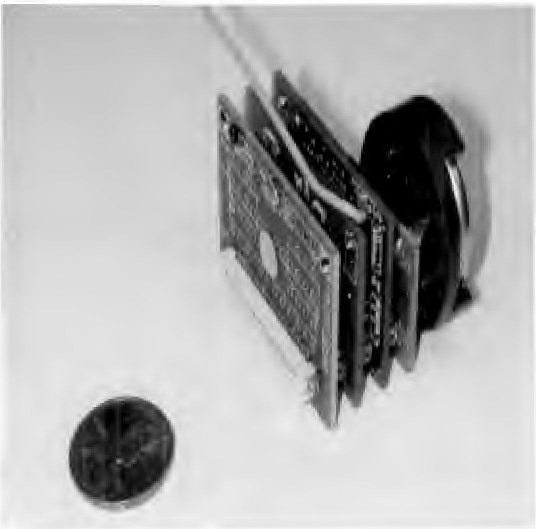


Figure 1.3: Rene Motes

Thus, depending on the phenomenon under observation, the measurements collected by sensor nodes may need to be subjected to proper processing which might be performed either in a distributed manner by nodes, or centrally where information is collected.

Distributed processing, or collaborative signal and information processing where nodes in an ad hoc wireless sensor network collaborate to collect data and process it into useful information, is a relatively new area of research [3]. Processing data from more sensors generally results in better performance but it also requires more communication resources (and thus energy). Energy efficient digital signal processors (DSPs) are becoming increasingly important with the growth of portable, wireless, battery-operated appliances such as wireless sensor networks, cellular phones, Personal Digital Assistants (PDAs) and laptops.

1.2. Fast Fourier Computing

Energy efficient digital signal processors (DSP's) are becoming increasingly important with the growth of portable, wireless, battery-operated appliances such as wireless sensor networks, cellular phones, Personal Digital Assistants (PDAs) and laptops.

The Discrete Fourier Transform (DFT) and the Inverse Discrete Fourier Transform (IDFT) are computational tools that play a very important role in many digital signal processing applications such as frequency analysis (spectrum analysis) of signals, power spectrum estimation and linear filtering. The formula for transforming for a sequence $\{x(n)\}$ of signals of length **L ≤ N** into a sequence of frequency of samples $\{X(k)\}$ of length N is called the Discrete Fourier Transform (DFT) and it is given by [3] :

$$X(k) = \sum_{n=o}^{N-1} x(n)\, e^{-j2\pi kn/N} \quad , \qquad (1)$$

where n=0, 1,..., N-1.

The formula that allows us to recover the sequences of $x(n)$ from the frequency samples is called the Inverse Discrete Fourier Transform (IDFT) and it is given by:

$$x(n) = \frac{1}{N} \sum_{n=o}^{N-1} X(k)\, e^{j2\pi kn/N} \qquad (2)$$

where n = 0, 1,..., N-1 .

The importance of the DFT and IDFT in such practical applications is due to the existence of computationally efficient algorithms, known collectively as Fast Fourier Transforms (FFT) algorithms for computing DFT and IDFT.

The FFT has been widely used in digital communication systems to optimize the applications of various parameters such as area and speed. The Cooley-Tukey algorithm is the most common FFT algorithms. It recursively decomposes a DFT of arbitrary length $N = N_1N_2$ into DFTs of smaller length N1 and N2. An important computational component of the FFT algorithm is the butterfly which takes inputs and performs computations that include complex number addition and multiplication. The radix-2 and radix-4 are some of the widely used FFT algorithm due to the simple structure of the butterfly components and periodic nature of the algorithm. This complex number multiplication and twiddle factors presents a challenge in most DSP applications in terms of energy and time consumption.

In [4] the idea of a Distributed Digital Signal Processor (DDSP is considered, where a network is composed of multiple sensor nodes, each of which corresponds to a processor. The idea of divide-and conquer approach is then applied, in which a DFT of size N, where a composite number is reduced to the computation of smaller DFTs from which the larger DFT is computed.

When implementing the FFT algorithms, in fixed-point number, the word length is a very important factor to be considered since it greatly affects both the performance and complexity of FFT processor. The processor with larger word length provides better performance in terms of accuracy. The accuracy of FFT processor is represented by Signal to Noise ratio (SNR), and a processor design with higher SNR value is desirable. On the other hand, a high word length in FFT design contributes to higher switching activities (SA). To resolve this problem, in [5] a single objective and a Multi-Objective Genetic Algorithm Approach is proposed to search for better SNR and SA value of the FFT processor.

1.3 Problem Statement and Research Questions

The problem statement and proposed research questions are discussed below.

1.3.1 Problem Statement

One of the most promising applications that are currently being studied for sensor networks is collaborative signal processing. The task of developing and implementing such pervasive applications still poses a tremendous challenge. In this context, the design of algorithms for processing information in sensor networks is an emerging research area.

Consider a group of sensors s_1, s_2, ..., s_k monitoring a single target. The target generates analogue signals and the sensors sense this information and encode it into digital data and subsequently send the data to a sink node. The problem is how do we minimize the energy utilized by the sensors to extend the life-time of the sensor network? This problem is further exacerbated by the limited communication range of radio communication used by the sensor nodes.

For most practical implementations, multi-hop transmission is necessary. Literature reveals that the main source of energy consumption in wireless sensor networks is signal transmission and reception, which consumes 50 times more than the energy required to process 1000 instructions [6, 7]. This means that the more hops we have the more the number of transmissions we have and the more energy consumed. Now consider a situation where we must send large amounts of data. This means we must break it down into smaller sizes that can be accommodated by the channel and send them over the several hops, resulting in a lot of energy consumption.

To minimize the amount of data transmitted, and hence energy consumed, it would be imperative to process the data first to efficiently process information stored in the distributed sensors. The processing of the data in the distributed sensors is further hampered by lack of a central entity for organization and control such as a base station as in other networks like the cellular system or the Internet [8]. Thus, this research explored how sensors in a distributed environment can communicate and process data before it is transmitted to the sink node.

The research also focused on the design of models for energy consumption distribution for multi-hop wireless sensor networks. Most of the previous analytical studies show that although advanced collaborative signal processing algorithms have been adapted, they do not capture the stochastic nature of sensing. Increasingly sophisticated wireless sensor networks with limited energy resources call for comprehensive cross-layer analysis of multi-hop networks.

To achieve high degrees of reliability in such networks, statistical information about energy consumption is required [9]. However, traditional energy approaches only focus on the average energy consumed. Thus, in the light of this gap, there was need for stochastic analysis of the energy consumption in a random network environment that leads to the development of a comprehensive cross-layer framework. Models realized from such frameworks were then used to investigate the relationships between the distribution of energy consumption and network parameters such as network topology, network density, duty cycle, and traffic rate.

The distribution of energy consumption was also used to investigate the node life cycle and network life cycle.

Another problem is that sensor nodes are deployed and monitor the environment for a long period and most of the times there would be little and no activity of interest, yet energy continues to be utilized. This research explored how signal compression techniques could be applied to minimize the amount data sent so as to reduce energy consumption.

Another problem that occurs is that neighboring sensor nodes tend to sense related signal information. The transmission of unprocessed related information also leads to energy wastage [9]. This research explored ways in which correlated data could be processed to minimize energy consumption.

Thus, this research sought find out the best methods to encode the sensed information, how the sensors should collaborate to process this information, thereby reducing noise, interference and redundancy before subsequently sending this information to the sink note through a routing process.

There are various approaches to resolve the problems above but as we revealed in the literature review, each has its own limitation. Attempts to resolve these problems using Fast Fourier Transforms have resulted in solutions that either introduced redundancy in complex number processing, lacked load balancing in complex number processing allocated to the sensors, or introduced additional constraints in memory requirements and switching costs. Thus, distributed signal processing using Fast Fourier Transforms remains a research issue of major concern to many researchers. Genetic algorithms apply when the elements are real, discrete or complex values. Thus, genetic algorithms are suitable for digital signal processing and Fast Fourier Transform computations.

1.3.2 Research Questions

The research explored the following research questions to address the research goal and objectives:

1. What are the existing technologies that are useful in distributed sensor networks to reducing energy consumption?

2. Can we employ the knowledge of Genetic Algorithms to develop energy efficient algorithms that would enhance energy consumption in sensor networks?

3. How can we develop a model for energy consumption distribution for multi-hop wireless sensor networks based on stochastic processes?

4. Is it possible to implement the developed algorithms to reduce computational complexity during signal processing in wireless sensor networks?

1.4 Research Goal and Objectives

The research goal and objectives are discussed below.

1.4.1 Research Goal

The main goal of this research was to investigate the use of multi-objective genetic algorithms in computing Fast Fourier Transforms for sensors networks to reduce redundancy in processing operations, data duplication during data transmission to enhance load balancing, signal compression and hence minimise energy consumption.

1.4.2 Research Objectives

To achieve the research goal, we focused on the following objectives:

a) To survey existing literature on the application of genetic algorithms in relation to computing Fast Fourier Transforms.
b) To develop energy efficient algorithms for computing Fast Fourier Transforms that reduce redundancy in processing operations, data duplication during data transmissions while achieving load balancing and overall energy conservation for the sensor network.
c) Develop a model for energy consumption distribution for multi-hop wireless sensor networks based on stochastic processes.
d) As proof of the concept, it was necessary to:

i. Implement the developed models and algorithms using simulation tools.
ii. Justify how sensors collaboration and distributed processing lead to energy conservation.
iii. Justify how genetic algorithms can be useful to obtain the best solution for computing FFT leading to energy conservation.

1.5 Rationale of the Study and Research Contributions

The rationale of the study and contributions of this research are discussed below.

1.5.1 Rationale of the Study

Distributed processing has the inherent benefit of speeding up data processing due to concurrent processing. There is also less communication bandwidth requirements because processing is located nearer the source of information. It also leads to more reliability because of lack of a single point of failure and improved responsiveness due to sensing, processing and source encoding being located at the source nodes. The benefit of distributed processing is that energy and computational limitations of individual sensor nodes can be overcome. For a, given level of processing complexity, the energy efficiency of the overall network could be increased significantly.

1.5.2 Research Contributions

The main contribution of this research to knowledge, academia and the research community is on:

i. The development of a Multi-Objective Genetic Algorithm based on field programmable gate arrays to resolve the challenge of complex number multiplication associated with the computation of Fast Fourier Transform during signal processing at the node level.

ii. The design of a novel stochastic context-aware energy consumption and distribution model capable of resolving the problem of load balancing in wireless sensor networks.

1.6 Research Methodology

While conducting this research, the following research methodology was used.

1.6.1 Literature Survey

Literature survey was carried out to identify how we could employ the knowledge of genetic algorithms to develop energy efficient algorithms that would enhance energy consumption in sensor networks. The literature survey also contributed in identifying important metrics for the development of models for Sensor Collaboration, distributed processing and energy consumption distribution for multi-hop wireless sensor networks based on stochastic processes.

1.6.2 Model Formulation

i. Multi-Objective genetic algorithms were employed to formulate a model for Multi-Objective optimization using FPGAs.

ii. A model for energy consumption distribution for multi-hop wireless sensor networks was constructed based on stochastic processes. The model is used to investigate the relationships between the distribution of energy consumption and other network parameters such as network topology, network density, duty cycle and traffic rate.

1.6.3 Evaluation and Implementation

As proof of the concept, we developed and simulated a Multi-Objective Genetic Algorithm based on Field Programmable Gate Arrays (MOGAFPGA) and a Stochastic Context-aware Energy Consumption and Distribution Model. The models developed were evaluated for precision, functional correctness and energy efficiency among other parameters. The models were also compared with other known standard protocols.

1.7 Included and Related Publications

We published the core chapters of our research work in peer reviewed journals and in this section we present the published works and those that have been accepted for publication but not yet published.

i. Njini, I. and Ekabua, O.O., 2014. Multi-Objective Genetic Algorithms for computing Fast Fourier Transform for evolving Smart Sensors devices using Field Programmable gate arrays. Advances in Computer Science: An

International Journal, 3(2), pp.1-9. This paper is part of chapter one, three and chapter five of this thesis. The work was also presented At North-West, University, Mafikeng Campus, Faculty of Agriculture, Science and Technology, Research Day, 08 October 2013.

ii. Njini, I. and Obeten O.O.Ekabua, 2014. Genetic Algorithm Based Energy Efficient Optimization Strategies in Wireless Sensor Networks: A Survey. ACSIJ Advances in Computer Science: An International Journal, 3(5), pp.1-9. This paper forms part of our Related Literature, Chapter two.

iii. Njini, I. and Obeten O.O.Ekabua, 2015.Stochastic Context Aware Energy Consumption and Distribution Model for Multi-hop wireless sensor networks ' was accepted for publication in ACSIJ Volume 4, Issue 2, March 2015. This article forms part of our Chapter four, five and six.

1.8 Thesis Structure

This thesis is structurally organized as follows:

Chapter 2 presents a detailed literature survey on Genetic Algorithms and Multi-Objective Genetic Algorithms. We present an analysis of various Multi-Objective based Optimization Strategies for energy efficient WSNs. We also present an analysis of various load balancing strategies for energy efficient WSNs. This same chapter also presents an evaluation of energy efficient techniques for correlated data in WSNs. The chapter ultimately focuses on evaluating the applicability of Distributed Source Coding (DSC) and Compressive Sensing techniques in the transmission of correlated signal data.

Chapter 3 presents development and simulation of a Multi-Objective Genetic Algorithm based on Field Programmable Gate Arrays (MOGAFPGA).

Chapter 4 presents the development and simulation of a Stochastic Context-aware Energy Consumption and Distribution Model (SCEDM) for WSNs.

Chapter 5 presents evaluation, results and implications for the research conducted.

Chapter 6 presents the summary of the thesis, conclusion and future work that could be based on this research.

Chapter 2

Review of Related Literature

2.1 Chapter Overview

In this chapter, we discuss research challenges in WSNs and present related literature on genetic algorithms, evolutionary algorithms, Energy Efficient clustering using GAs, Multi-objective genetic algorithm based optimization problems and strategies, load balancing, genetic algorithms in modelling sensor radio models, link quality-based routing protocols, GAs based load balancing approaches, energy efficient techniques for WSNs and approaches for transmitting correlated data in WSNs such as Distributed Source Coding, compressive sensing and Geographic routing. We also review signal reconstruction methods such as greedy iterative algorithms, convex relations, iterative thresholding algorithms and Bregman iterative Algorithms. We also review FFT, applications of FFT in distributed processing, GAs in Computing FFT, low power FFT architecture, Field programmable Arrays, context aware computing and stochastic modelling process.

2.2 Research Challenges in WSNs

The past decade has witnessed tremendous growth in research in various issues of concern in wireless sensor networks (WSNs) such as energy conservation, node deployment, routing protocols, Quality of services (QoS) management, security, energy harvesting etc. Most of the issues involved in WSNs research are conflicting in nature and hence require optimization strategies that are capable of mitigating the conflicting objectives such as life time maximization, node coverage and reliability among others.

Genetic algorithms apply when the elements are real discrete or complex valued. Thus, genetic algorithms are suitable for digital signal processing and fast Fourier transform computations. As has been discussed earlier in Chapter 1, wireless sensor networks are composed of hundreds or thousands of sensor nodes deployed in the environment for purposes of detecting and transmitting information of interest. According to Chen et.al., [10], a wireless sensor network (WSN) is a collection of wireless sensor nodes forming a temporary network without the aid of any established infrastructure or centralised administration. In such an environment, there is limited range therefore it is necessary for one sensor node to collaborate for the help of another node in forwarding packets to its base station. Usually the device nodes consist of CPU for data processing, memory for data storage, battery for energy and a transceiver for receiving and sending signals from one node to another. WSN are generally characterized by short range radio communications, limited computational capacity and limited generally irreplaceable battery power [11].

One of the major concerns in the design and deployment of wireless sensor networks is maximizing network life time. The problem is how do we minimize the energy utilized by sensors in order to extend the life of the sensor network? Network life span is dependent on network connectivity and network connectivity in turn depends on network bottleneck nodes [12]. This problem is further exacerbated by the limited communication range of radio communication used by the sensor nodes. For most practical implementations, multi-hop transmission is thus necessary.

In multi-hop wireless sensor networks, nodes that are close to sink node transmit much more data, and then exhaust their energy while other nodes in the same network remain with energy. In [13], it is estimated that the energy consumed in transmitting *k* bits of data over a distance *d* is:

$$E(k,d) = E_{elec} * k + E_{amp} * k * d^2 \tag{3}$$

Where E_{elec} is the radio energy dissipation and E_{amp} is transmitting amplifier energy dissipation. Now consider a situation where we have to send a large chunk of data, and *d* is the distance between communicating nodes and *k* is a constant of proportionality. This means we have to break it down into smaller chunk sizes that can be accommodated by the channel capacity and send them over the several hops discussed above, resulting in a lot of energy consumption. It was therefore necessary to research into the various ways in which energy can be conserved as sensor nodes transmit data from source to sink within a WSN. Literature revealed that genetic algorithms play an increasingly important role in the design and deployment of wireless sensor networks (WSNs). Recent advances in wireless sensor networks have led to many new routing protocols and clustering methods using genetic algorithms specifically designed for energy awareness.

2.3 Genetic Algorithms

Genetic algorithms are efficient stochastic optimization search procedures that mimic the adaptive evolution process of natural systems. They have been successfully applied in many NP-hard problems such as multiprocessor deign, task scheduling, optimization and travelling salesman problem. Genetic algorithms are most useful in problems with large irregular search space where a global optimum is required. Traditional gradient based methods of optimizing generally encounter problems when the search space is multimodal because they tend to become stuck at local maxima. Genetic algorithms tend to suffer less from this problem of premature convergence [14].

A genetic algorithm is an iterative approach, involving trial and error, which aims to find a global optimum. Nature's equivalent is the process of evolution over time, where many members are created, and each population becomes better adapted to its environment. We may simulate an evolution process by creating an initial pool of chromosomes, where each chromosome represents a typical solution to the problem we intend to solve and taking the following steps illustrated in figure 2.1 and the algorithm bellow [15]:

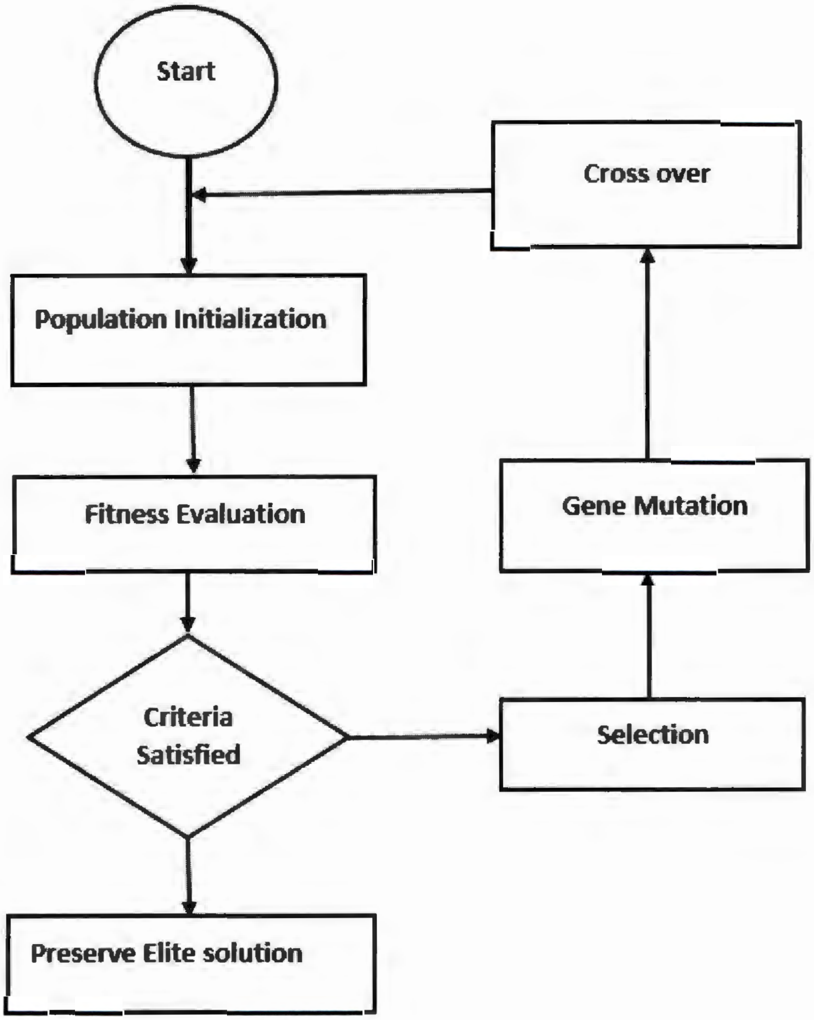


Figure 2.1: Genetic Algorithm

1. Create a random population of N chromosomes (Candidate solutions for the population).
2. Evaluate the fitness function $f(x)$ of each chromosome x in the population. Generate a new population by repeating the following steps until the new population reaches population N:
3. Select two parent chromosomes from the population, giving preference to the fitter chromosomes (high $f(x)$ values). Automatically copy the fittest chromosome to the next generation (this is called 'elitism').

4. With a given crossover probability, cross over the parent chromosomes to form two new offspring (if no crossover was performed, offspring is exact copy of the parents).
5. With a given mutation probability, randomly swap two genes in the offspring.
6. Copy the newly generated population over the existing population.
7. If the loop termination condition is satisfied, return the best solution in current population.
8. Otherwise go to Step 2.

We generally let this process go on for a predetermined number of generations, or until the standard deviation of the fitness converges towards zero (when the standard deviation starts to converge, the chromosomes are generally getting fitter, so we have arrived at the best solution we can find). If the initial population is large enough, and the fitness is well defined, we would have arrived at a good solution [16].

Genetic algorithms do not find the best solution or the ideal solution. However, if we run a simulated evolution many times, they tend to find a very good solution. So how does this process evolve fitter genes? Some of the evolutionary spirals that move towards fitness come from mutations that introduce new gene sequences to the population, but most Genetic Algorithm success comes from crossover. By combining portions of fit chromosomes in new ways through random crossover, Genetic algorithms evolve over time evolve to become even fitter chromosomes [17].

2.4 Research Considerations in Evolutionary Algorithms

When implementing evolutionary algorithms (EAs), it is necessary to specify the method of parent selection, crossover, mutation and control parameters such as population size and number of generations. These are briefly discussed below.

2.4.1 Parent Selection Schemes

The selection process is probabilistic in nature but in practice one needs a method for identifying good parents to select for mating in order to produce the next generation. The following parental selection schemes have been predominantly used in the implementation of evolutionary algorithms:

Proportionate reproduction: In this scheme, individuals are chosen for birth in proportion to their fitness value. The probability that an individual from the i^{th} class (having common fitness value f_i is chosen for selection in the t^{th} generation is [18]:

$$m_{it} = \frac{f(x_i)}{\sum_{i=1}^{n} f(x_i)} \qquad (2)$$

where m_{it} is the number of individuals in the population at time t with fitness i. Proportionate reproduction is usually implemented with a Monte Carlo or roulette wheel selection.

Ranking Selection

In ranking selection the population is selected from best to worst. The number of copies that an individual should receive is given an assignment function, and it is proportional to the rank assignment of an individual.

Tournament selection: In tournament selection, a random number of individuals is chosen from the population (with or without replacement) and the best individual from the group is chosen as a parent for the next generation. This process is repeated until the mating pool is filled. There are a variety of other selection methods, including stochastic remainder and universal selection [19].

2.4.2 Encoding

The encoding process can be viewed as a mapping from a list of individuals in the possible solution space to strings of codes in the model or Algorithm, generally referred to as chromosomes. Encoding of chromosomes is a very important question to ask when solving a problem with genetic algorithms. Encoding depends heavily on the problem. There are many ways of encoding and these depend on the problem to be solved. Binary encoding is the most common one, mainly because the first research on genetic algorithms used this type of encoding and its relative simplicity. In binary encoding, every chromosome is a string of bits, either 0 or 1. A chromosome should, in some way, contain information about the solution that it represents [20].

2.4.3 Cross Over, Mutation and Repair

Selected chromosomes go through processes of crossover and mutation. Crossover is a recombination process whereby two selected parent chromosomes swap certain gene components thereby evolving the characteristics of the resultant off-spring individuals. The crossover operation is usually applied with high probability of occurrence. The mutation process randomly changes particular string positions and this has low probability of occurrence. Sometimes the crossover between two valid chromosomes may lead to an invalid offspring. The identification and exclusion of invalid chromosomes is achieved using repair functions.

2.4.4 Fitness Evaluation

In natural selection, the fitness of an individual may be regarded as the ability to pass genetic material to the next generation and hence the ability for an individual's quality to survive. In GAs, the fitness of a chromosome is a function that can be evaluated for each chromosome to determine its ability to solve the problem at hand. In general chromosomes with higher values have better chances of survival.

Section 3.0 discusses binary encoding for cluster based problems and section 3.1 discusses binary encoding for routing problems in wireless sensor networks

2.5 Clustering

Clustering partitions the network into groups of sensor nodes which are geographically close to each other. Each cluster has a cluster head which is responsible for controlling all the activities of the group such as transmission, aggregation, management and maintaining structure. With clustering in WSNs, energy consumption, lifetime of a network and scalability can be improved. Currently, the accepted and mostly used topology for clustering in WSNs is where each cluster has a cluster head. The sensor nodes transfer their data directly to their associated cluster head nodes (relay nodes) and then cluster head nodes perform the initial data aggregation and send it to the designated route [21].

2.5.1 Energy Efficient Clustering using Genetic Algorithms

Genetic algorithms apply when the elements are real discrete or complex valued. Thus, genetic algorithms are suitable for digital signal processing and fast Fourier transform computations. Genetic algorithms partition the sensor network into independent clusters using GA to minimize the total communication distance, and thus prolong the lifetime of a network [22]. In [23], an intelligent clustering approach in wireless sensor networks is proposed. The approach uses a genetic algorithm to minimize the total communication distance by using a binary representation in which a bit corresponds to a sensor node. A "1" corresponds to a cluster-header, while a "0" corresponds to an ordinary sensor node as depicted in figure 2.2 and 2.3. Consider the following example, representing a typical chromosome, or GA solution.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	0	0	0	1	0	0	1	0	0

Figure 2.2: Typical Chromosome representation

Nodes S1, S5, S8 are the cluster headers and the rest are ordinary nodes. Two parent chromosomes, parent 1 and parent 2 representing two different clustering solutions can then be engaged in a process called crossover to generate two new solutions, child 1 and child 2 as indicated in Figured 2.4.

Parent1

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	0	0	0	1	0	0	1	0	0

Parent 2

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
0	0	1	0	0	1	0	1	0	0

Figure 2.3: Parent Chromosomes used for crossover

Child 1

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
0	0	1	0	1	0	0	1	0	0

Child2

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	0	0	0	0	1	0	1	0	0

Figure 2.4: Evolved Child chromosomes

This implies that initially we had S1, S5 and S8 as the cluster headers for one typical solution, and S3, S6 and S8 as cluster headers for another solution. After the crossover, two solutions are generated with S3, S5 and S8 as the new cluster headers in one solution set and S1, S6 and S8 as the cluster headers in another solution set. After a new population is generated the performance of the clusters generated was evaluated using a fitness function metric which depends on the total distance of nodes to the sink, and a weighted ratio of the number of cluster heads to ordinary nodes. The limitation of the approach by Jin et:al [23] is that this model does not take into consideration the fact that sensor nodes may alternative between sleeping mode and active modes.

2.5.2 Cluster Based Optimisation Strategies

An Energy Efficient Clustering Scheme Based on Grid Optimisation using a genetic algorithm which drives the network area into virtual grids with each grid representing a cluster is presented in [24]. The genetic algorithm is used to divide nodes equally among the grids to ensure load balancing and thus enhancing the network lifetime. The model does not consider the fact that energy consumption patterns vary as one moves from source to sink, with nodes closer to sink transmitting more, and hence spending more energy. In [25], an algorithm which does cluster head selection using fuzzy logic and chaotic based genetic algorithm based on fussy logic is presented. Each node calculates its chance based on its energy, density and centrality. Nodes that have high energy inform the base station as potential cluster header candidates. The base station uses the genetic algorithm based on chaotic reasoning to select the cluster headers. Although this approach tries to use information on residual energy as well as node density and centrality to ensure prolonged network life time, it suffers the drawback that this increases communication between the sensor nodes and the base station is another form of energy wastage. This scenario is illustrated in Figure 2.5.

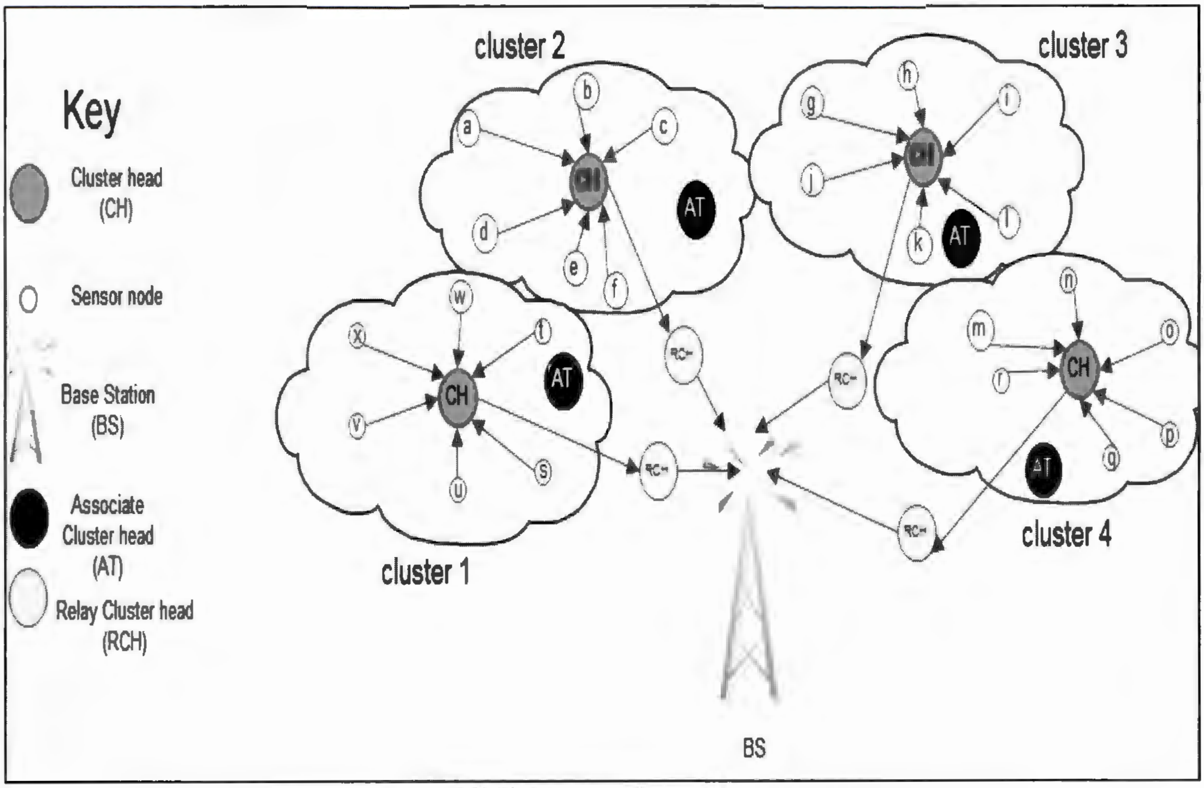


Figure 2.5: Network configuration

In [26] a Genetic Algorithm Based Energy Efficiency Clusters (GABEEC) algorithm is proposed. This approach consists of two phases, the set-up phase and the steady-state phase. In the set-up phase the cluster heads are selected and other nodes are assigned to the cluster heads as ordinary nodes based on distances as depicted in Figure 2.6. In the steady-state phase, the nodes send their data to the cluster heads which in turn send forward it to the base station. The Base station then checks the energy levels of nodes and CH, and if a cluster head is having low energy, then an associate CH is selected from the population. This selection is done using Roulette-wheel selection method.

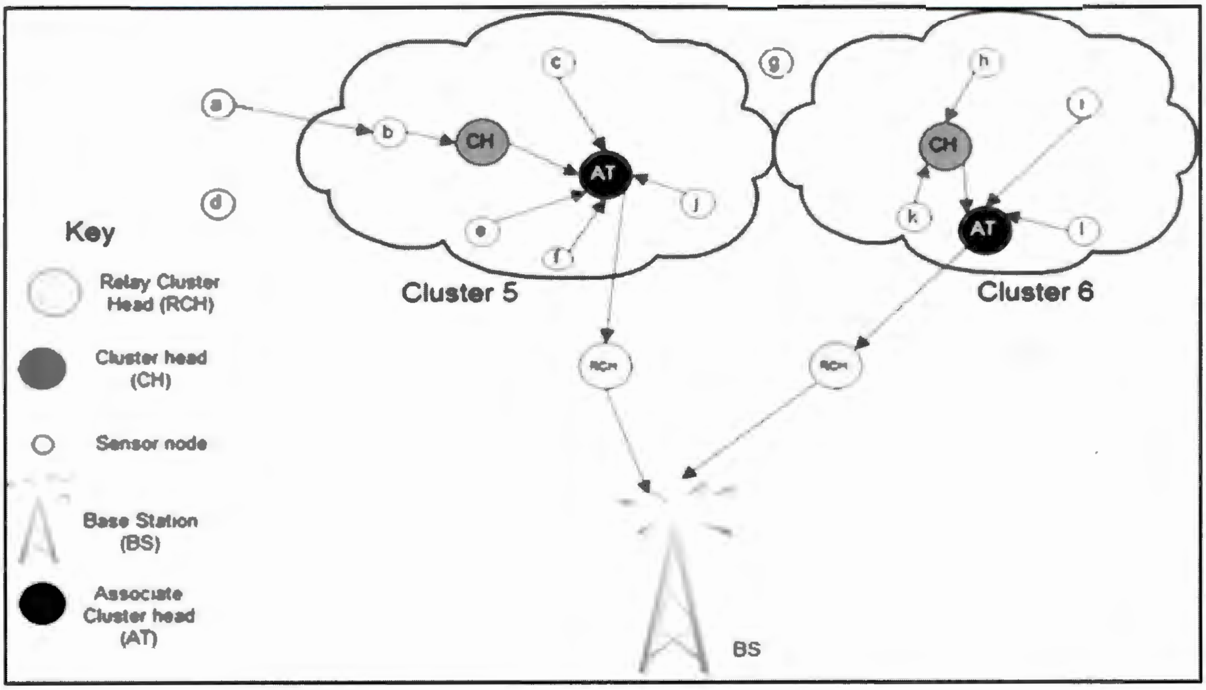


Figure 2.6: Network configuration in the Steady-state phase

While this approach attempts to maximize the network life time by minimizing communication distance, it also increases communication overheads in order to send information about the residual energy to base station. This increased communication leads to depletion of energy, hence reducing the network life time.

2.6 Multi-Objective Genetic Algorithm based Optimisation Strategies.

The broad application of wireless sensor networks has resulted in the development of a wide variety of techniques which are NP hard and most of them difficult to obtain high precision solutions by traditional methods. Thus, while employing genetic algorithms to solve problems in WSNs, it is important to form a broad review of the current research and future trends in the use of genetic algorithms and multi-objective genetic algorithms in particular WSNs. The characteristics of wireless sensors networks determine a different kind of design problem with different requirements for detailed applications.

There is need for good routing protocols that should make comprehensive considerations of multiple factors to satisfy the transmission requirements of different data with Quality of Services (QoS) parameters that may be conflicting and different such as end-to-end delay, energy efficient routing, node placement and layout optimization etc.

2.6.1 Multi-Objective Optimisation Problem

The general multi-objective optimisation can be modeled as

$$\min\ y = f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

$$subject\ to\colon \{x = (x_1, x_2, \dots, x_n) \in \mathcal{F}_x \subset X \qquad (4)$$

$where\ x \in X$ is called the decision vector, X is the optimization space, $f \in Y$ is the objective vector, Y is the objective space, and the set $\mathcal{F}_x$ is the feasible set composed of solutions which satisfy the problem constraints. Let the vector f be described component-wise by $\boldsymbol{f} = (f_1, f_2, \dots, f_m)$, and let $\mathcal{F}_y \subset Y$ represent the image set of region $\mathcal{F}_x$ for the mapping $\boldsymbol{f}(.)\colon \boldsymbol{X} \mapsto \boldsymbol{Y}$ [27]. The set of solutions of a multi-objective problem consists of all decision vectors in which the corresponding objective vector can be improved in any dimension without degradation in another one.

This set of solutions is known as the Pareto-Optimal set. Each element of the Pareto-Optimal set constitutes a non-inferior solution to the multi-objective optimization problem. The problem has usually no unique, perfect solution, but a set of equally efficient, non-inferior, alterative solutions (Pareto-optimal set). Each point in this set is optimal in the sense that no improvement can be achieved in one vector component that does not lead to degradation in at least one of the remaining components. The set of non-dominated solutions lie on a surface known as the Pareto-Optimal frontier [28].

In most cases, there will be several optimal solutions in the Pareto sense and we must look to the values of the objective function in order to decide which values seem most appropriate.

2.6.2 Multi-Objective Optimisation Algorithms Approaches

A multi-objective optimization algorithm is one that deals directly with a vector objective function and seeks to find multiple solutions offering different, optimal tradeoffs of multiple objectives. There are basically three approaches to tackling multi-objective optimization problems which are as follows [29]:

1. Ignore some of the attributes entirely and just optimize one that looks most important.
2. Lump all attributes together by just adding them up or multiplying them together and then optimize the resulting function.
3. Apply a multi-objective algorithm that seeks to find all the solutions that are non-dominated. Non-dominated solutions are those that are optimal under any reasonable way of combining the different objective functions into one. A non-dominated individual is one where an improvement in one objective results in deterioration in one or more of the other objectives when compared with the other individuals in the population.

Thus in this paper we argue that (3) seeking multiple, distinct solutions to a problem, conferring different tradeoffs of objectives, is the essence of true multi-objective optimization (MOO). In the next section we discuss various implementations of MOO in WSNs.

2.6.2.1 Hybrid Multi-Objective Optimisation Strategies

In [30] a Dynamic Multi-objective Hybrid Approach for designing WSNs is presented. The approach proposes a multi-objective hybrid approach for solving Dynamic Coverage and Connectivity problems (DCCP) is a network with no cluster heads and with nodes subject to node failures. The rationale behind this approach is to maximize the network life time by minimizing the number of active nodes in each time period, while complying with the network requirements. The presented approach using a local online algorithm intended to restore the network coverage when one or more node failures occur. The limitation of this approach lies in the assumption that there are online resources that may be available to a WSNs and this may not always be the case.

2.6.2.2 Quality of Service Routing

In [31] Ekbatanifard et:al, proposed a Multi-Objective Genetic Algorithm based Approach for Energy Efficient Qos-Routing in Two-tiered Wireless sensor Networks.

The approach optimizes the network by routing data from source to sink in such a way that the three conflicting objectives, end-to-end delay, transmission reliability, and node residual energy are optimised. Thus, these QoS parameters are used to form a multi-objective function that serves as a performance criteria for identifying optimal routes. We briefly discuss how the Qos parameters can be modeled for clarity.

2.6.2.3 End to end Delay

Consider a network in which *n* source nodes transmit data to relay nodes which in term perform fusion and retransmit the data to the sink node through relay nodes via a multi-hop WSN. Then the networks of relay nodes form a routing tree, with sources, several intermediate nodes and a single sink node. The routing tree can be modeled as a graph [32] [33],

$G = (V, E)$, where V is the set of relay nodes and E is the set of edges. A path between source node (V_d) and relay node (V_r) can be represented as a sequence, v_r, v_1, v_2, $\ldots v_d$, where $v_i \in V$.

The delay over a particular route from source to sink follows a Weibull distribution with parameter μ . The Weibull distribution gives the probability distribution of lifetimes of objects. It was originally proposed to quantify fatigue data, but it is also used in analysis of systems involving a "weakest link." Thus, the Weibull distribution can be used to model devices with decreasing failure rate, constant failure rate, or increasing failure rate. This versatility is one reason for the wide use of the Weibull distribution in reliability.

The Erlang distribution can be used to model the time to complete *n* operations in series where each operation requires an exponential period of time to complete. The probability that a delay d_p over an individual path *k* is less than *t* is estimated by the Erlangen distribution,

$$P_r\,(d_p < t) = \frac{(\frac{\mu}{\alpha})^{k\beta}\, t^{k\beta-1} e^{-(\mu t/\alpha)\beta}}{(k-1)!} \qquad (5)$$

Where α >0 is a scale constant, and β is the shape parameter. The multi-objective function seeks to minimise this objective.

2.6.2.4 Reliability

Path reliability can be defined as the expected number of successful end-to-end forwarding transmissions (and retransmissions) of data for a successful end-to-end delivery of and hop-by-hop acknowledgement (ETX). For a path *p* consisting of links $v_1, \ldots, v_n$ with forward delivery ratio fd_{vi}, and reverse delivery ratio of rd_{vi} for link *vi*, the reliability metric EXT may be computed as:

$$etx_{vi} = \frac{1}{(fd_{vi} \times rd_{vi})}$$

$$ETX(p) = etx_{v1} + etx_{v2} + \cdots + etx_{vn} \tag{6}$$

The reliability of the entire routing tree is then computed by maximizing the whole routing tree reliability given by [34]:

$$R_{tree} = \left(\frac{\sum_{p \in Tree} EXT(p)}{\sum_{p \in Tree} 1}\right) - 1 \tag{7}$$

2.6.2.5 Energy

Energy consumption by a relay node may be estimated by the following equation:

$$E_T(b, d_{ij}) = \theta_1 b + \gamma b d_{ij}^m \tag{8}$$

$$E_R(b) = \theta_2 b$$

Where d_{ij} is the Euclidian distance between node *i* and *j*, θ_1 is the transmit energy coefficient, γ is the amplifier coefficient, *m* is the path loss exponent, $2 \leq m \leq 4$, and θ_2 is the received energy coefficient. *b* represents the traffic bit-rate in relay nodes which depends on current bandwidth.

2.6.2.2 Routing Algorithm based Optimisation

The approach by EkbataniFard et.al., [30] [35] then uses a genetic algorithm to find routes from source to sink that optimizes the QoS parameters, discussed above. An initial population is first constructed using depth first search algorithm. Using this population, an initial set of routing trees is constructed. Each of these routing trees is then encoded into chromosomes that represent typical routes, with each integer representing a sensor node. These chromosomes then participate in mating (crossover) to generate new routes in the network. Figure 2.7 shows how two typical parents participate in a crossover to generate two child chromosomes typifying the generation of new routes from combinations of old ones.

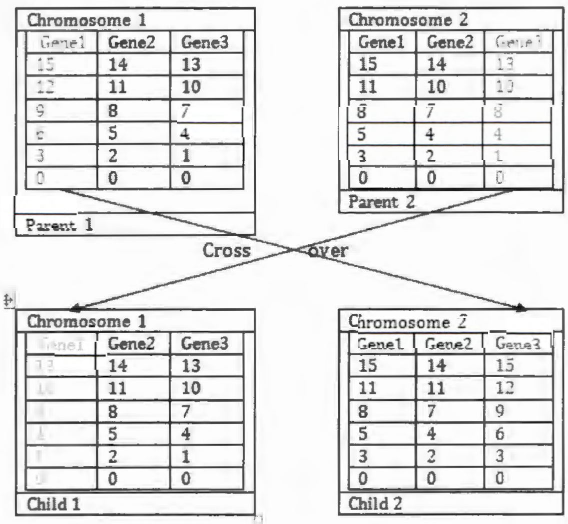


Figure 2.7: Crossover operation

With a given probability *p,* mutation is carried out on carefully chosen nodes to ensure feasibility of the new paths. A new population $P_t = P_t \cup Q_t$ is formed where *t* is the number of generation.

2.6.2.2 Multi-Objective Optimisation based on Routing and Flow rate

Mohamed [36] et.al., proposed a Multi-Objective Optimisation approach that jointly minimizes energy consumption and traffic delay. The approach derives an energy model based on the assumption that the energy consumed by a node is proportional to the amount of data transmitted over a link. Assuming pairs of connected nodes i-j, and denoting the set of all connected nodes denoted by *L*, the capacity of each link denoted by Q_{ij} may be defined as the maximum number of packets per second that can be transmitted over the link. The optimization problem is constrained by the flow rate equality constraint:

$$\boldsymbol{Ax} = \boldsymbol{b}, \tag{9}$$

where $\boldsymbol{A}_{(l,l')_k} \in \mathbb{R}^{mxn}$ denotes the path incidence matrix for source link pairs $\boldsymbol{l}, \boldsymbol{l}'$, $\boldsymbol{x}$ denotes path flow rate vector, and $\boldsymbol{b}$ is the rate at which data is generated at the source. The model is also constrained by the total flow $\boldsymbol{Y}$, which is the sum of all flows passing through the link. The total flow should not exceed the link capacity $\boldsymbol{Q}$. The total flow $\boldsymbol{Y}$ is denoted by:

$$Y = Cx, \tag{10}$$

where $\boldsymbol{C}$ is the link incidence matrix defined as:

$$(\boldsymbol{c}_{i,j})k = \begin{cases} 1, & if\ k \in F_{i,j,} \\ 0, & otherwise, \end{cases} \tag{11}$$

where $\boldsymbol{F}_{ij}$ denotes the collection of paths that include link $(\boldsymbol{i}, \boldsymbol{j})$.

To transmit data between two successive nodes, the energy consumed at a node is composed of transmission energy and reception energy. Transmission energy is a function of the distance between the sending node and the receiving node and the size of data flow. Reception energy is a function of the amount of data received. The transmission energy consumed at a node may be expressed as:

$$\boldsymbol{E}_t(\boldsymbol{y}_{ij}) = (\epsilon_t + \epsilon_{amp}\, \boldsymbol{d}_{ij}^2)\boldsymbol{y}_{ij}, \tag{13}$$

Where ϵ_t is a coefficient associated with energy transmission per bit per second for transmitting data, $E_t(y_{ij})$ is a function of that depends on the distance between the transmitter and the receiving node, and lastly ϵ_{amp} is the transmission amplifier coefficient. The reception energy may be expressed as:

$$\boldsymbol{E}_r(\boldsymbol{y}_{ij}) = \epsilon_r\, \boldsymbol{y}_{ij}, \tag{14}$$

where ϵ_r is a coefficient associated with energy consumption per bit per second for receiving data. The total energy consumed at each node is then expressed as:

$$E_i(x) = \epsilon_r Cx + \epsilon_t Cx + \epsilon_{amp} Cxdd^T, \tag{15}$$

The problem was formulated as follows:

$$Minimise \qquad E_{total}\,(\boldsymbol{X})$$

$$Minimise \qquad \Psi_{total}(\boldsymbol{X})$$

Subject to:

$$(c1) \qquad Ax = b$$

$$(c2) \qquad Cx \leq Q$$

$$(c3) \qquad \epsilon_r Cx + \epsilon_t Cx + \epsilon_{amp} Cxdd^T \leq E_i$$

$$(c4) \qquad \Psi(x) \leq D_{max}$$

$$(c5) \qquad x \geq 0.$$

2.7 Load balancing Strategies in WSNs.

Large scale applications of WSNs result in increased energy wastages due to problems associated with increased network activity that leads to packet collisions and node failures. It is therefore necessary to employ congestion control protocols that ensure efficient management of network load distribution to avoid problems associated with *hot spots* [37]. The problem of hot spots normally occurs in resource limited WSNs when routing protocols select nodes with favourable channels to relay traffic (including traffic that they did not even generate) resulting in overloading of the nodes compared to the other peer nodes. Hot spots depend to a large extent on the topology of the network.

Load balancing aims at avoiding the formation of hot spots. Congestion control protocols have traditionally been used to achieve balanced distribution of traffic among various possible paths that data packets pass through as they traverse from source to sink. Load balancing scenarios vary from simple cases where all nodes can only transmit data that they generate from source to sink in one hop, to cases in which there exists only a few particular relays of nodes capable of transmitting from source to sink in the network or subsection of the network, thereby creating topological bottlenecks [38].

A line network is a typical extreme case in which nodes can only receive data from their upstream neighbours and can only transmit to their downstream neighbours. Figure 2.8 Shows a line network depicting a topological bottleneck transmitting data from source A, through peer nodes B, C, D to sink S. In cases where the sensor network comprises of such line networks, it is not possible to avoid bottlenecks as nodes on the downstream get an ever-increasing work load than their upstream counterparts.

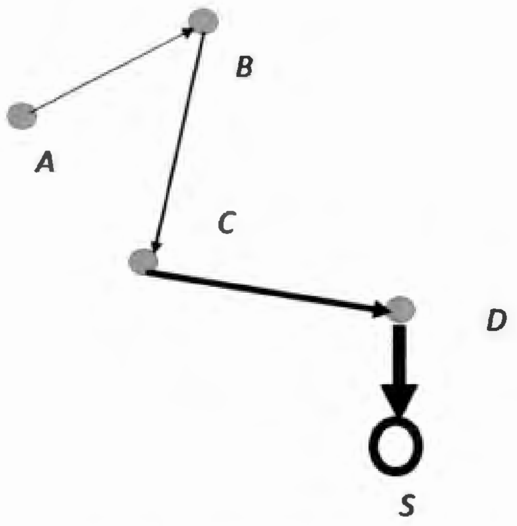


Figure 2.8 Topological Bottleneck

The rationale behind load balancing is that it averages the energy dissipation, which in turn extends the average time the first node runs out of energy, hence extending the expected lifespan of the WSN. We briefly discuss the various strategies for load balancing in WSNs and we evaluate the impact that they have on energy efficiency [39].

2.7.1 Routing Tree based load balancing.

In [40] a routing tree based load balancing approach is presented. The approach is based on initially constructing an unbalanced routing tree structure. The scheme later readjusts the routing tree using a rebalancing algorithm to achieve even distribution of the sensor nodes within the tree structure. Figure 2.9 depicts a typical unbalanced routing tree structure and Figure 2.10 shows a typical balanced routing tree structure.

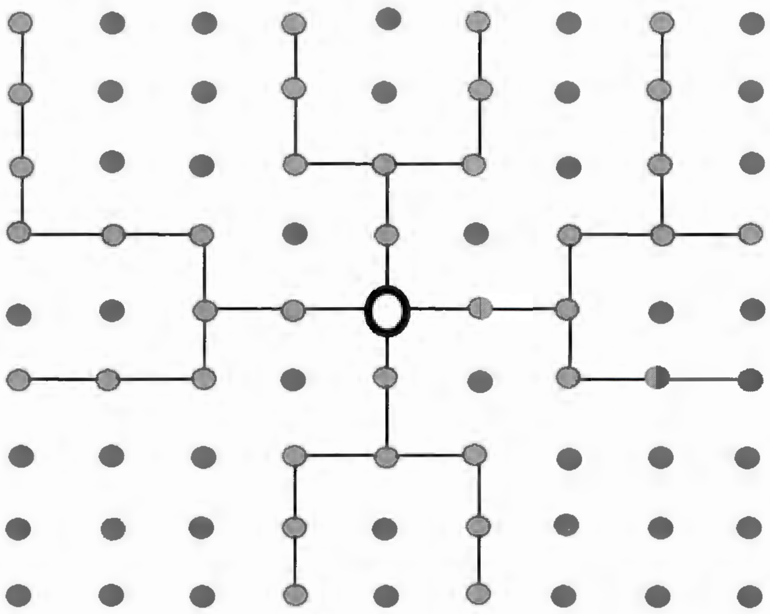


Figure 2.9: Unbalanced Tree Structure

In [41] a Node Centric Load Balancing Algorithm is reported which is based on initially constructing a generally node based routing tree on a square grid. The scheme uses the concept of Chebyshev Sum Inequality to ensure cumulative node balancing within a hierarchical tree structure. A spiral adjustment algorithm which utilizes topology information is later introduced to further refine the node balancing.

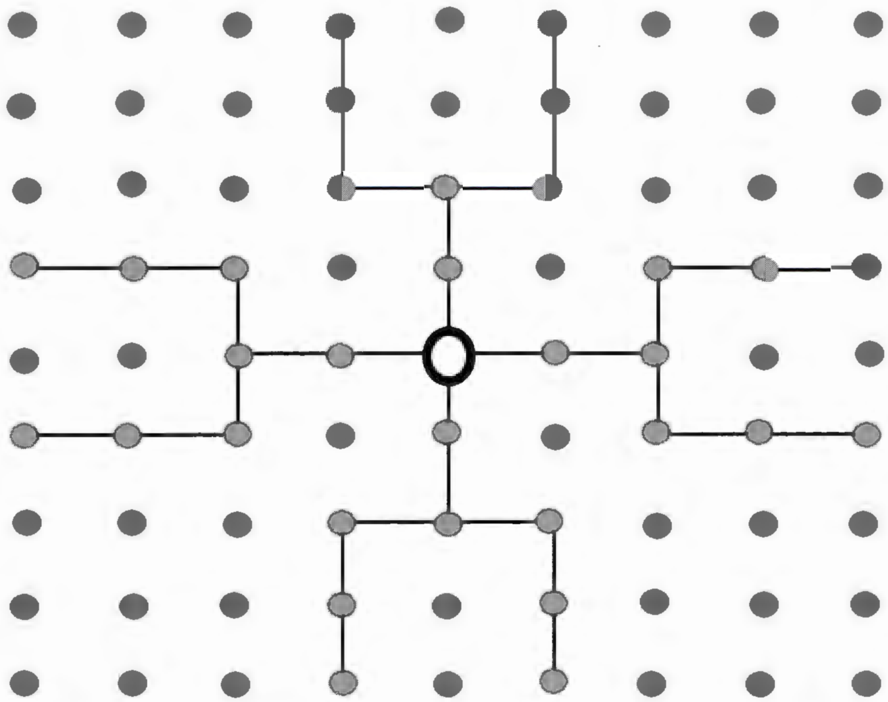


Figure 2.10: Balanced Tree Structure

The limitation behind tree based algorithms is that they tend to concentrate on the distribution of nodes and topology structure rather than real network load information. True load balancing should focus more on the distribution of network load due to network activities (sensing, transmissions and receptions) and not just distribution of nodes within a topology structure. The tree based node balancing structure does not account for differences in node activity as time progresses as some nodes tend to sense and transmit/receive more data than others due to differences in activities generated on different spatial locations. Thus, tree based strategies are prone to the creation of *hot spots*.

2.7.2 Link Quality-based Routing Protocols

Literature reveals that link estimation based routing protocols have been successfully applied in computing reliability-oriented route selection protocols. In [42] the route selection process is broken down into three components, namely Link Estimation, Routing Engine and a Forwarding Engine.

The Link Estimation Component is responsible for assessing the quality of the wireless links in the network. The Routing Engine is responsible for determining the next hop node to forward data based on the routing cost function in the equation. The metrics of the cost function are derived from Link Estimation information and other relevant routing data. The Routing Engine evaluates the best performing routes based on the cost function:

$$C_k^i = 2 - l_{(k,i)} - L_i + 2B_k, \tag{17}$$

where C_k^i denotes the cost incurred by node ***i*** if it routes data through ***k*** $B_k = max_{j=0..h-1}\beta_{p^j}(k)$ defines the bottleneck load factor ***k***, and $L_k = min_{j=0..h-1}l(_{p^j(k),p^{j+1}(k)})$ denotes bottleneck link quality over a multi-hop route with ***h*** hops.

While MultiHopLQI [43] picks the most reliable links, it does not explicitly pursue load balancing. There is no deliberate effort to distribute link usage in such a manner that the load is balanced as it is relayed from source to sink. The technique only achieves limited load balancing in the presence of fading as link quality values are modified during parent selection. The approach exploits suboptimal links to balance between reliability and load balancing factors.

Evaluation of load balancing protocols may be done using traditional metrics such as average hop count and maximum hop count. A more interesting metric is the relaying cost-to-benefit ratio (η) which estimates the life time of the network as a ratio of relayed load to delivered load.

$$\eta = \frac{\sum_{i\epsilon N}\beta_i}{\sum_{i\epsilon \mathcal{R}}\Pi_i} \tag{18}$$

where, N is the set of all nodes, $\Re$ is a subset of all relay nodes and Π_i represents packet delivery ratio (PDR) for node *i* to sink.

2.7.3 GA based Load Balancing Approaches

Literature reveals that Connected Dominating Sets (CDS) [44] in combination with genetic algorithms and other biologically inspired algorithms have been used to try and resolve the problem of load balancing in WSNs. In this section, we review some of the inspiring and innovative approaches.

In [45] Genetic algorithms are used for achieving load balancing in a connected dominating set problem. The approach initially constructs a Virtual Backbone using connected dominating sets. The idea of a connected dominating set has been used in literature to construct virtual backbone (VB) to achieve efficient routing in WSNs. A Dominating set for a WSN is a subset of the nodes in the network such that each node is in the set or adjacent to a node in the set. The network is thus partitioned into dominators (nodes in the CDS) and dominatees (other nodes). Dominatees forward data to neighbouring dominators and only dominators relay the data through the CDS. Figure 2.11 shows a typical CDS that is not load balanced. The dominator nodes are denoted by the black circles while the white circles denote the dominatees' nodes. The initial allocation of dominatees to dominators is represented by solid black lines while dotted lines indicate alternative communication between the dominators and dominatees. Let the set of neighbouring dominatees nodes allocated to dominator d_i be denoted by $S(v_i) = \{ v_j | v_j \in W, d_j$ forwards its data only to $v_i\}$. The virtual back bone is B= {v_4, v_7} and the MCDS that can be constructed would be $S(v_4) = \{ v_1, v_2, v_3, v_5, v_6\}$ which implies that dominator v_4 connects to 5 different dominatees while on the contrary $S(v_7) = \{ v_6, v_8 \}$ only connects to two dominatees. Assuming all nodes to be generating and transmitting equal amounts of data, then this implies that node v_4 will quickly deplete its energy and create a hot spot faster than v_7 would. Furthermore, dominate, v_6 connects to two nodes v_4 and v_7. If it were to be connected to v_4 only then load balancing would be further compromised as only one dominatee v_8 would be forwarding data to dominator v_7.

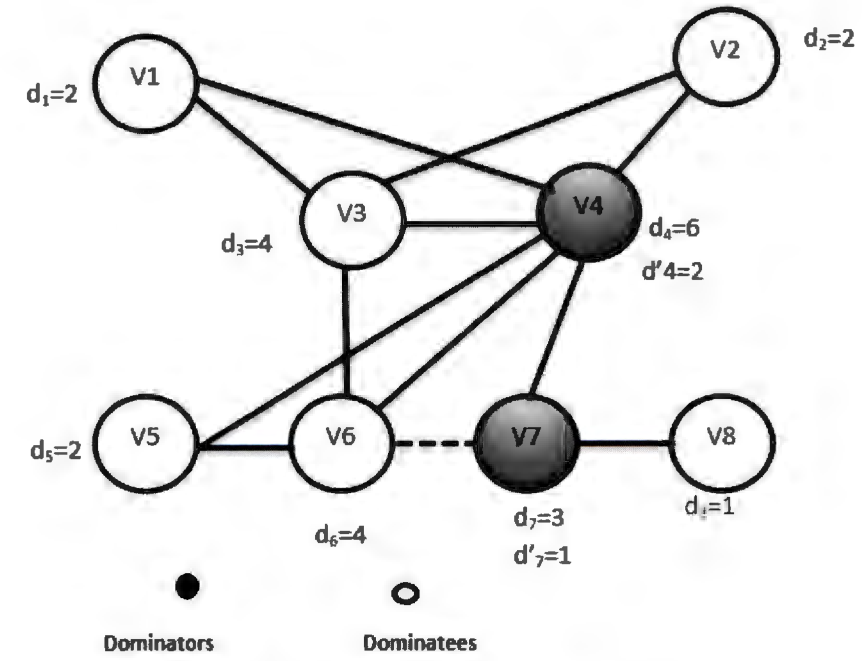


Figure 2.11: CDS with no load balancing

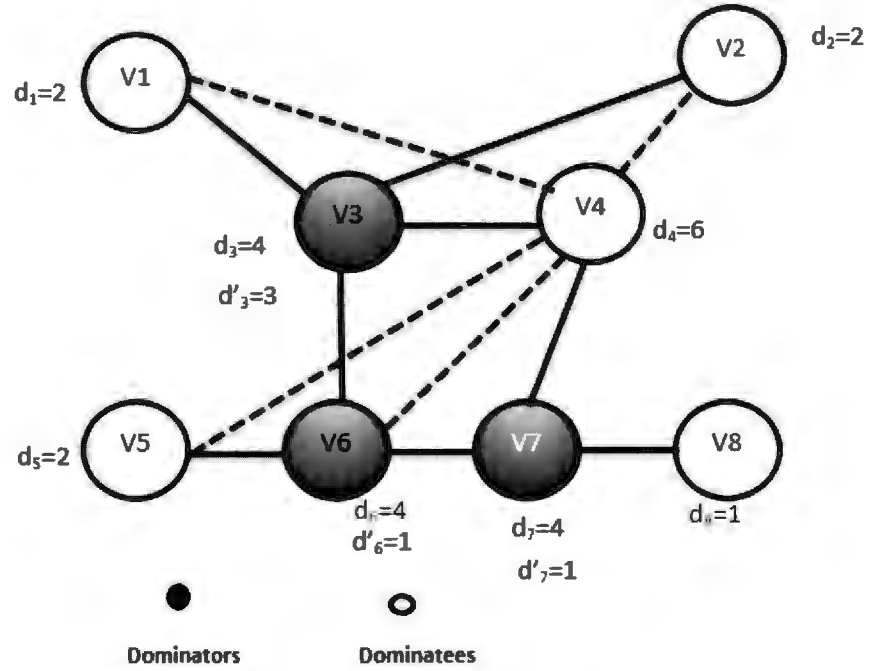


Figure 2.12: CDS with load balancing

Figure 2.12 and 2.13 shows counter examples to the problems of unbalanced load discussed above and illustrated in Figure 2.11. In Figure 2.12 and 2.13, the virtual backbone is the set B= {{v_3, v_6, v_7}. The connected dominating sets for the topology in Figure 2.12 are $S(v_3) = \{v_1, v_2, v_4\}$, $S(v_6) = \{v_4, v_5\}$, $S(v_7) = \{v_4, v_8\}$. In this case the CDS leads to a balanced distribution on network nodes which might, in turn, lead to load balancing assuming that the network load is generated uniformly. The same applies for Figure 2.13 with the following CDS $S(v_3) = \{v_1, v_2\}$, $S(v_6) = \{v_4, v_5\}$, $S(v_7) = \{v_8\}$.

In [46] the author proposed a *k*-connected *m*-dominated set as an abstraction model of a fault tolerant VB. In [47] a CDS based on minimum routing is proposed. The approach aims to minimize CDS while ensuring the shortest path through the CDS. While prior efforts to employ CDS to build minimum connected dominating sets have been partially successful, their full potential cannot be realized when used in isolation. Genetic algorithms have been used to ensure that the entire network is connected and load balanced by encoding a chromosome as a possible solution of the LBCDS problem. Thus, load balanced connected dominating set genetic algorithms (LBCDS-GA) have been proposed in literature.

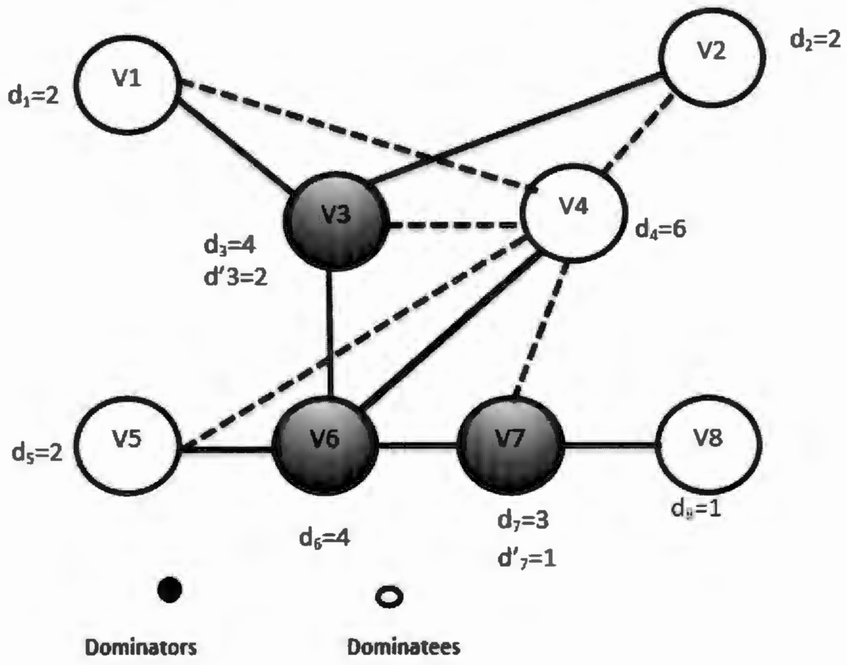


Figure 2.13: k-CDS with load balancing

The author in [48] proposed a Genetic Algorithm inspired by Load balancing protocol for Congestion Control in Wireless Sensor Networks using Trust based Routing Framework (GACCTR). The Approach uses encoded GA chromosomes to construct initial routes based on route trust. It then uses GA mutations to compute new routes from the initial population of routes.

2.8 Genetic Algorithms in Wireless Sensor Radio Models

In [49] biologically inspired Wireless genetic algorithm is reported. The model uses information about the radio channel in the physical layer to construct a statistical model based on external channel environment as depicted in Figure 2.14. The WCGA then passes channel parameter statistics to a Cognitive Monitoring System which incorporates this information into chromosomes and then computes the necessary genetic algorithm computations such as selection, crossover and mutation leading to different radio configurations. A Cognitive System Monitor is then used to incorporate capabilities of learning and adaptation into different radio models. Finally, another component, a Wireless System Genetic algorithm, is used to compute the necessary fitness function to evolve the radio configurations.

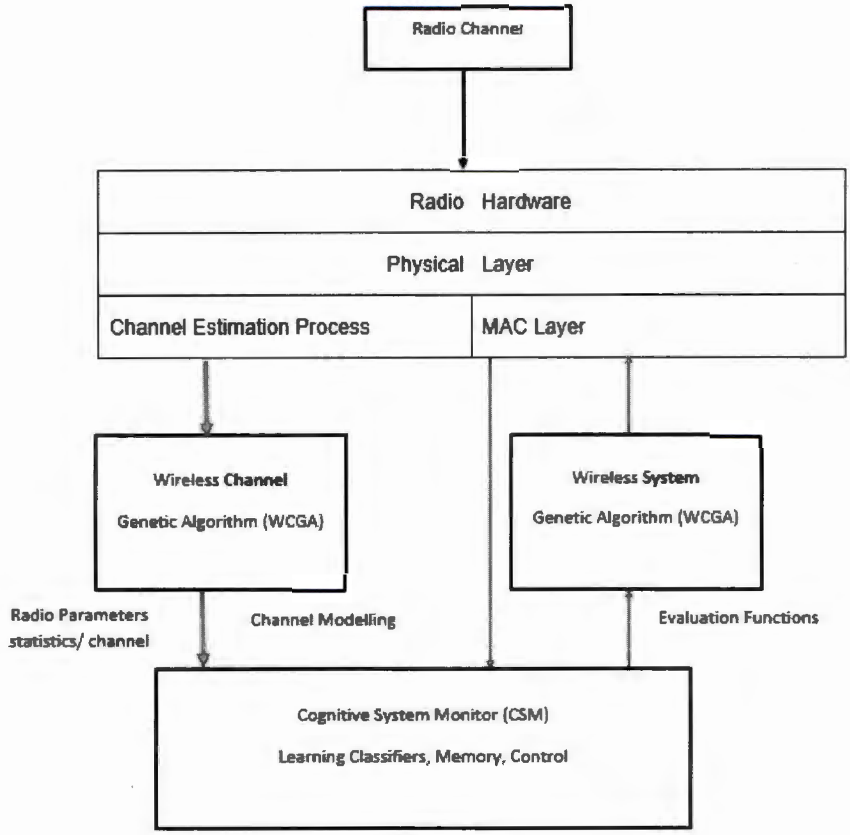


Figure 2.14: Cognitive Engine Architecture

2.9 Nature of Correlations in WSNs

Wireless sensor networks (WSNs) are usually densely deployed to measure phenomenon that generate observations that are highly correlated in nature such as temperature, humidity etc. When deployed in this manner, the sensor readings tend to be generally correlated. The transmission of this redundant data leads to energy wastage as more data packets are transmitted and received than is necessary. Moreover, this increased transmission may lead to increased packet collisions and hence data loss or other network related problems. However, the correlation of sensor data can be exploited in order to come up with energy efficient protocols for WSNs [50].

Literature reveals that a substantial amount of work has been done to design WSNs protocols aimed at exploiting the correlation structure to achieve energy efficiency, but there is need for comparative evaluation of the suitability of the work carried out this far in meeting the various requirements of distributed wireless sensor networks. To the best of our knowledge, there are no research works that provide a holistic evaluation of the suitability of recent energy efficient techniques for exploiting correlations in WSNs.

The main purpose of this research is to provide further understanding of the research community by categorizing, summarising and comparing recent research techniques and methods for exploiting spatio-temporal correlations in WSNs. We seek to explore the characteristics of spatio-temporal correlations and establish the extent to which they have been exploited towards achieving energy efficiency in WSNs protocols.

WSNs correlations can generally be categorized into two components, namely spatial correlation and temporal correlations. *Spatial Correlation* is a result of multiple sensor nodes deployed close to each other in space, observing and transmitting records about the same event. The degree of this spatial correlation is reported to increase with decreasing distance between nodes [51].

Temporal Correlation is a result of a WSN system periodically performing observation of events and transmitting these observations. Temporal correlation measures the degree of similarity between consecutive observations of a sensor node and this varies with the temporal characteristics of the phenomenon under observation. In Event tracking applications, it might be necessary to adjust the frequency of reporting and hence transmission frequency in order to capture temporal correlation phenomenon at acceptable distortion levels and at minimum energy consumption of the WSN application.

Thus, the goal in most WSNs research is to have the largest network coverage that maintains the highest possible levels of spatial and temporal resolution. To achieve high levels of spatial resolution, one must package the concerned area with a significant density of nodes. On the other hand, temporal resolution is achieved through sampling the environment at a high frequency.

2.9 Approaches for Communication WSNs Correlated Data

In the next section, we briefly discuss approaches that are targeted at reducing both spatial correlations and temporal correlations. The techniques that we discuss are Opportunistic Routing, Compressive Sensing, Distributed Source Coding and Transform coding.

2.9.1 Opportunistic Routing Approaches

Due to the limited communication distance of WSN radio, it is sometimes necessary to broadcast data and retransmit it through other relay nodes until it reaches the destination. In the absence of appropriate protocols for sensor co-ordination, duplicated retransmission may occur, leading to increased energy depletion. Moreover, the link conditions in WSNs are constrained by factors such as interference, attenuation, and channel fading, making it a challenging problem to provide reliable and timely communications in WSNs. To enable reliable packet forwarding of data, it may be necessary to retransmit packets at each hop, but this may then result in undesirable energy wastage and delay.

Several existing works [52], [53] have proposed the use of multiple paths between the source and the sink. Although these approaches have managed to achieve the desired QoS reliability and delay, they however suffer from the disadvantage that sending a packet over multiple paths introduces significant energy costs which is a major design concern for WSNs. Moreover, sending data over multiple channels increases channel contention and interference leading to delivery delay or even transmission failure in some instances.

Routing protocols are responsible for discovering and selecting the paths that data traverses from source to sink. We can differentiate these into two paradigms, namely traditional routing and opportunistic routing, depending on when the route path is discovered and selected. Opportunistic routing is based on the discovery and selection of route paths prior to the transmission processes while opportunistic routing is a proactive and reactive approach in which the route path is discovered on demand and periodically [54].

Opportunistic routing (OR) is a new technique that aims to manage the coordination of the process of retransmission in order to avoid the duplication of transmissions. It is based on the notion that when data is broadcast by the source node, other intermediate nodes may overhear the transmission and coordinate amongst themselves to elect forwarders that then relay the data until it eventually reaches the destination. In OR [55], the set of nodes that are elected as next hop forwarders are known as the candidate set C_i. In OR, any of the candidate nodes may retransmit the data, resulting in potentially different paths through which the data may be routed. Thus, by coordinating among the candidates that successfully received the transmitted data, an appropriate forwarder may be selected to achieve transmission reliability and network throughput.

Geographic opportunistic routing (GOR) [56] is an emerging area of routing that aims to improve the network performance of WSNs by exploiting spatial diversity of dense networks achieved through using location information available at each node. In GOR [57] QoS constrained geographic opportunistic routing protocol is presented for multi-hop WSNs. The approach uses two attributes, namely single hop packet progression and packet reception ratio to construct a set of forwarding candidates, F_i. Single hop packet progression is based on the Euclidian distance between the potential candidate node and the destination node. The packet progression ratio is defined as:

$$a_{ij} = Dist(i, Dest) - Dist(j, Dest), \tag{19}$$

where $Dist(i, Dest)$ denotes the Euclidian distance between node i and the sink node. The Packet Reception Ratio (PRR) is denoted by p_{ij}. Each node i that intends to send data to the sink node maintains the pair of information (a_{ij}, p_{ij}) for each neighbor j in a table. When a node intends to send a data packet to a destination node, it initially selects a forwarding candidate set F_i from the list of available next hop candidates set C_i based on its local information, where $F_i \subseteq C_i$.

The node then broadcasts a data packet which comprises of the data to be transmitted, the list of forwarding candidates and their respective priorities appended in the data headers. When the forwarding candidates successfully receive the packet, they each start a counter, with a duration that is inversely proportional to their priority ranking, with those that have higher priority setting shorter timers while those with lower priority ranking set longer timers. A candidate whose timer expires first sends an Acknowledgment (ACK) reply notifying the sender and other candidates to abort their timers. The candidate thus becomes the next hop forwarder and relays the data to the next hop node. The receiving node relays the data in a similar manner and the process continues to repeat itself until the data reaches its destination. The EQGor algorithm below shows how it selects a candidate node for relaying data to the next hop nodes until it finally reaches the destination. The algorithm principally select forwarding candidates from a set C and sort them in descending order according to a metric derived as a product of their advancement and their packet reception ratio. The algorithm uses two parameters α and β to which determine the minimum and maximum number of forwarding candidates.

Input: The available next-hop node set C(C≥ 2, partitioned hop Qos requirement: speed, r_i.

Output

The forwarding candidate set F

$F = \{C_1\}; k = \max\{\alpha, \min\{\beta, 20\%|C|\}\};$

while $C \neq \emptyset$ **do**

If *Constraints meet,* **then**

return F;

else if *CheckRange (F,C1)*

==false **then**

//node c1 is always the first node in C so it should be within the transmission range of F;

$C_i \leftarrow C_i - \{c_1\};$

Continue;

else if $|F_i| \leq k$ **then**

$C_i = C_i - \{c_1\}$

for i=0 **to** |F| **do**

Temporarily insertc_1 between F(i) and F(i+1)

end

Insert (c_1, i^*, F);

else Append (c_1, F);

$C = C - \{c_1\}$;

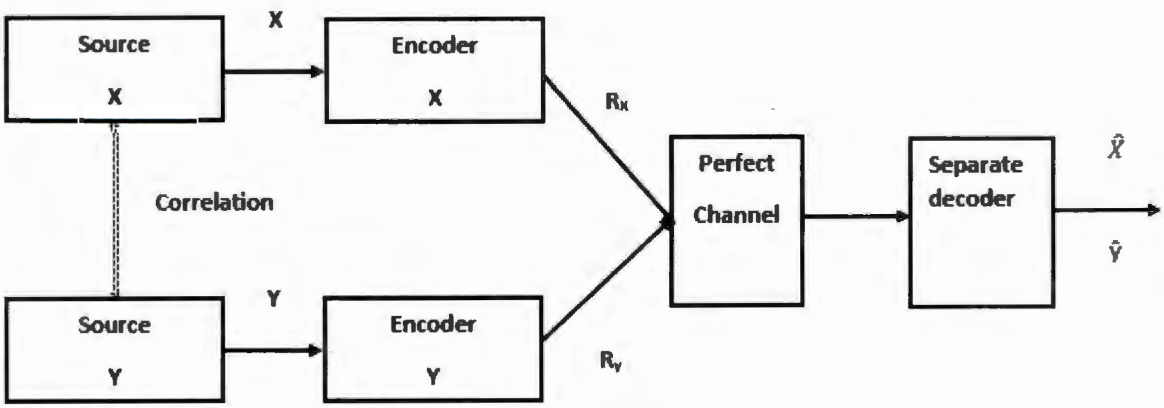
end

end

EQGor is constrained by inaccuracy associated with the metric used for selecting Forwarding Candidates. The metric does not consider the residual energy of selected forwarding candidates; hence it may result in the creation of bottlenecks.

2.9.2 Distributed Source Coding

The Slepian-Wolf coding problem presents a promising Distributed Source Coding (DSC) technique which is capable of resolving data redundancy resulting from the transmission of spatially distributed sensors in a WSN [58].



2.15: Separate Encoding Scheme

The Slepian-Wolf theorem states that two or more dependent non-communicating sources can be encoded separately and decoded jointly, as depicted in figure 2.15 at the rate at which they are jointly coded without any performance loss, provided the minimum data rate for each source is bounded by [59] :

$R_x + R_y \geq H(X, Y),$ (20)

$R_X > H(X/Y),$

$R_y > H(Y/X).$

Given a homogenous WSN with sensor readings denoted by:$X = X_1, X_2, X_3, \ldots, X_N,$ and that the spatial correlation structure has the probability distribution $p\,(x_1, x_2, x_3, \ldots, x_n)$.

Assume that the WSN can be partitioned into various sets V representing cluster within the network, such that $V \subseteq \{1,2, \ldots, N\}$, denotes sets of nodes in a WSN, and V^c is a complementary set of V, then the Slepian-Wolf coding theorem shows that if there are two (or more) data sources that are correlated, the results of encoding them independently are the same as with that of mutually encoding them. Hence nodes can encode with their joint bits) are bigger than their joint entropy [59] :

$R(V) \geq H(X(V)|X(V^c)$ (21)

$X(V) = \{X_j | j \in V\}.$

According to the chain theory, it is possible to find a rate allocation vector $\{R_i\}_{i=1}^{N}$, such that the number of generated bits from all nodes can achieve the value of their joint entropy.

An extension of the Slepian-Wolf problem is the Clustered Slepian-Wolf coding problem [60] which considers the problem of choosing a set of non-overlapping potential clusters to cover an entire network in such a way that the overall compression gain of Slepian-Wolf is maximized. For a network with N nodes, reading values X_i, using shortest path with weight $w(i, \hat{S})$, transmitting to sink node $\hat{S}$, the global rate allocation problem is finding the optimal rate, vector $\{R_i^*\}_{i=1}^{N}$for all nodes N, such that the total flow cost $\sum_{i=1}^{N} e(i,s) R_i^*$ is minimized under the following constraints: $R_1^* = H(X_1)$

$R_i^* = H(X_i | X_{i-1}, X_{i-2}, \ldots, X_1)$, (22)

$2 \leq i \leq N$, where $e(i, s)$ denotes the weight associated with the shortest path from i to sink S. To resolve this problem [61] Pu Wang et.al., proposed a distributed optimal compression clustering protocol. The approach builds a solution based on locating an optimal intra-cluster rate allocation based on Slepian-Wolf coding within clusters.

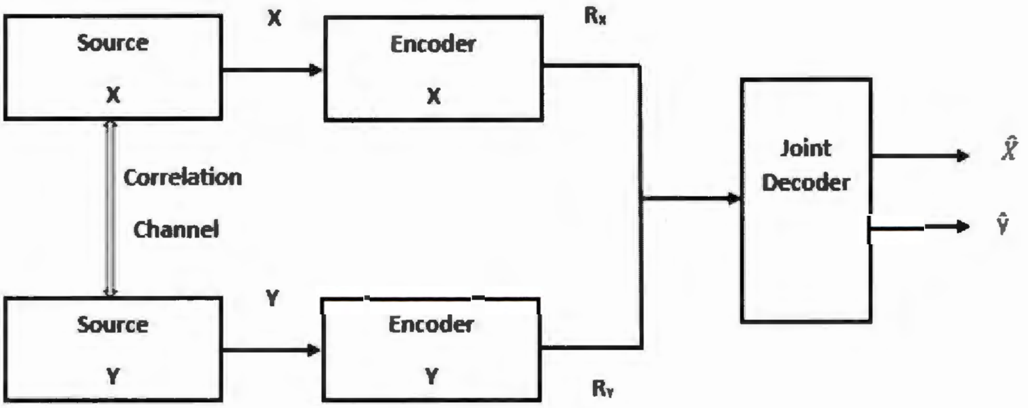


Figure 2.16: Distributed Compression of two correlated channels

The author in [62] proposed a joint decoding scheme illustrated in figure 2.16 which is reported as suitable for low power transmission and hence suitable for sensor networks with limited resources. However, the problem with this scheme is that spatial correlation and temporal correlation are discussed separately.

The author in [63] considers the design of a joint iterative encoding scheme which caters for both spatial and temporal correlation. The proposed decoder gains additional coding by exploiting the spatial and temporal correlation of two binary sequences of data. In the approach, two turbo decoders are joined with each other, thereby exchanging side information in an iterative way. At each decoding iteration, spatial and temporal information is extracted from side information and feedback to the turbo decoders as additional *a priori* information. The approach also utilizes a buffer to store side information which is used for decoding future sequences.

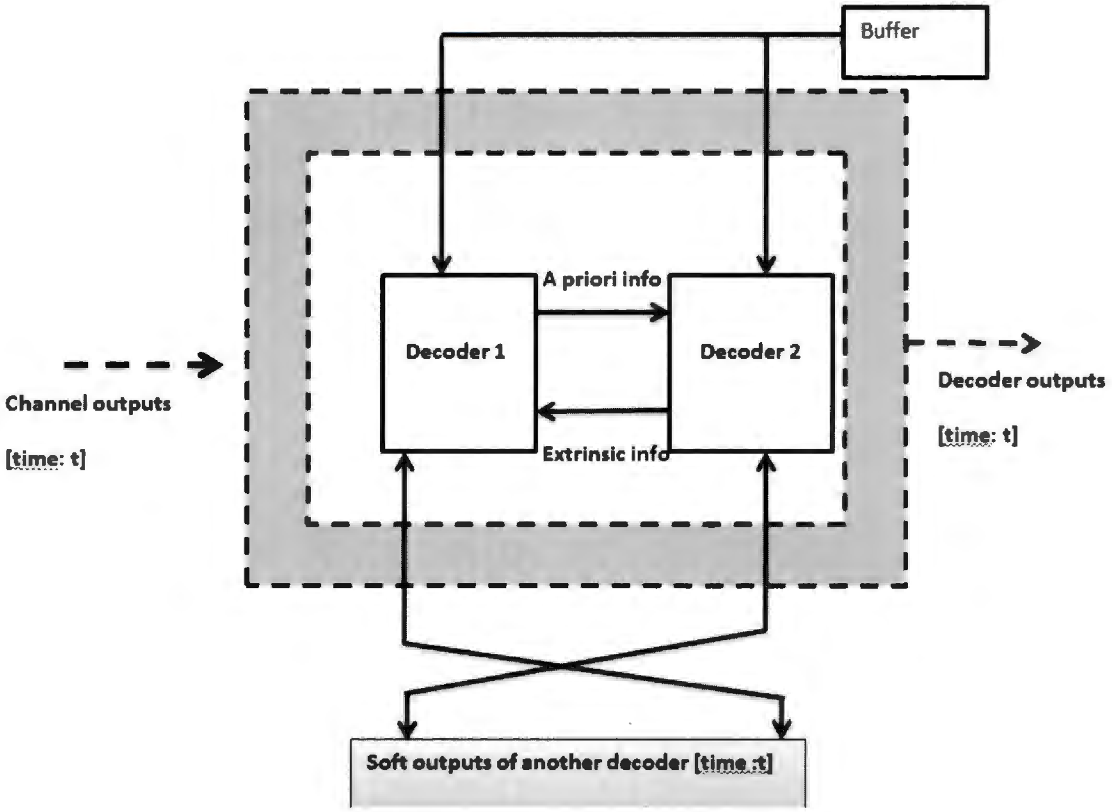


Figure 2.17: Joint Iterative Decoding Scheme

Figure 2.17 shows a block diagram of the Joint Iterative Decoding Scheme. Akyildiz [64] presented a theoretical framework for developing spatio-temporal models. The source *S* in a sensor field generates signals which are encoded by an encoder *E* and the transmitted signal eventually arrives at the Sink *S1*. The Sink is interested in receiving the observations with a high degree of accuracy. The physical process under observations maybe represented as a spatio-temporal process s(t,x,y), where t denotes time and (x,y) denotes coordinates of a point in space.

Depending of the specific implementation of the WSN application, the major interest might be to reconstruct the original signal at the sink node based on the observation of various sensor nodes, while in other applications the intention would be to reconstruct the signal at multiple locations. Observed sample readings $X_i[n]$ of each sensor node n_i, over a time period n was represented as [65] :

$$X_i[n] = S_i[n] + N_i[n] \tag{23}$$

Where *i* denotes the spatial location of node n_i , and $S_i[n]$ denotes the realization of space-time process $S(t,x,y)$ at any given time $t = t_n^1$ and location $(x,y) = (x_i,y_i)$ and $N_i[n]$ is the observation noise. The sensor observation ***Xi[n]*** passes through a source coding process to generate encoded information *Yi[n]* according to the formula $Y_i[n] = f_i(X_i[n])$. Figure 2.18 shows a conceptual illustration of spatio-temporal correlations. The sink node generates an estimate of the original signal $\hat{S}$. The sink node aims to reconstruct the source node *S* in a manner that minimizes the distortion according the constraint:

$$D = E[d(S,\hat{S})] \qquad (24)$$

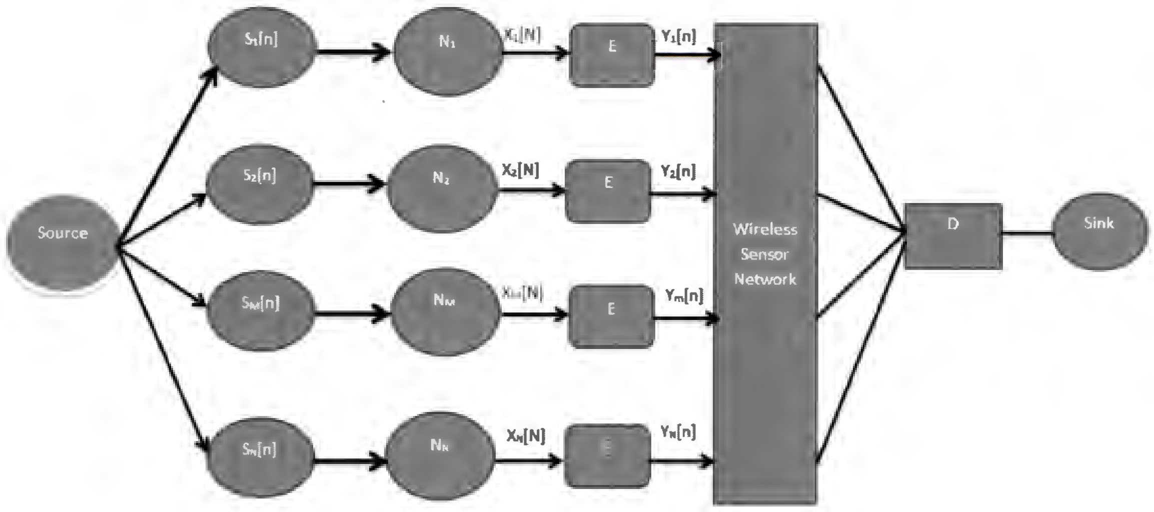


Figure 2.18: Spatio-temporal Correlation

The covariance function, $K_{\vartheta}(.)$ models the relation between the correlation coefficient between the sensor observations $\rho_{(i,j)}$, and the distance, $d_{(i,j)}$, between the nodes n_i, n_j. The power exponential was used to model the covariance function $K_{\vartheta}(.)$, as follows:

$K_{\vartheta}^{PE}(d) = e^{(-d/\theta_1)}$; for $\theta_1 > 0$,. (25)

The approach uses the distortion achieved by transmitting fewer numbers of packets *M*, to estimate an event source *S* using the distortion function:

$$D[M] = \sigma_S^2 - \frac{\sigma_S^4}{M(\sigma_S^2+\sigma_N^2)}(2\sum_{i=1}^{M}\rho(s,i) - 1) + \frac{\sigma_S^6}{M^2(\sigma_S^2+\sigma_N^2)^2}\sum_{i=1}^{M}\sum_{j\neq i}^{M}\rho(i,j). \quad (26)$$

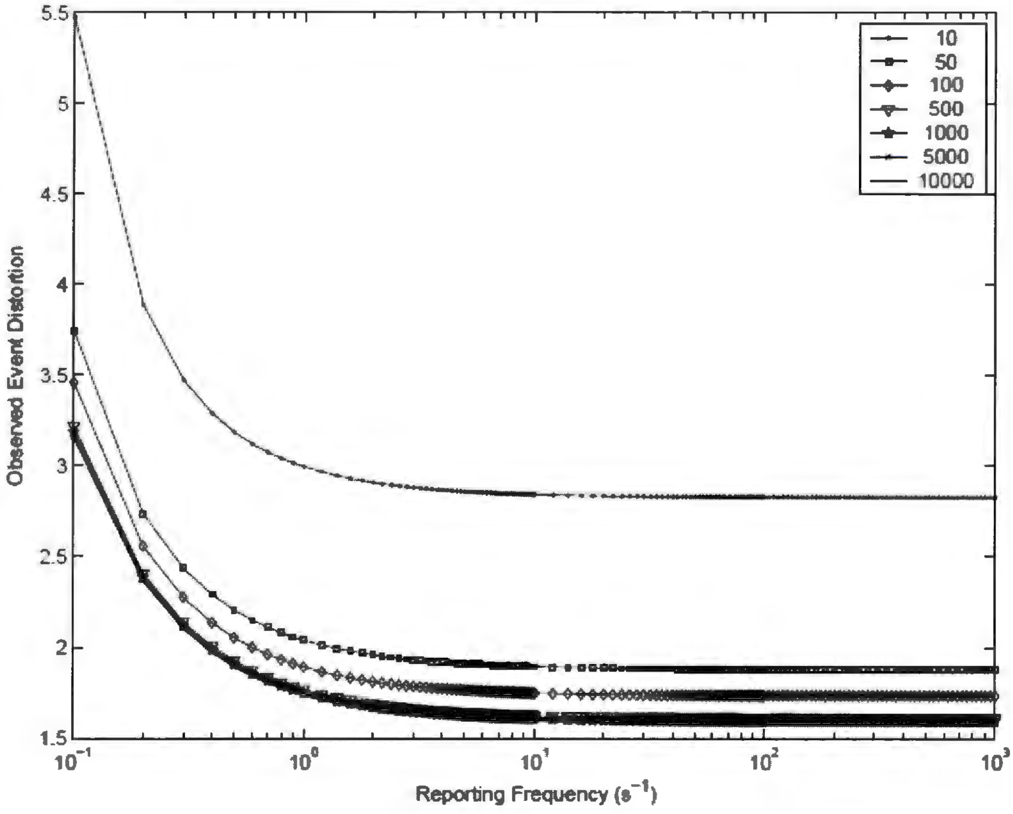


Figure 2.19 Observed Event Distorting VS Reporting Frequency

Figure 2.19 shows that Observed Event distortion inversely varies with increase in reporting rate. In [66] a Multi-rate Distributed source coding approach is used and it is reported to have reduced energy consumption for targeted BER values. The approach used Low Density Parity check codes within a DSC framework to achieve near Slepian Source coding limits. In [67] a data-adaptive sampling technique uses prior observations to focus feature measurements.

2.9.3 Compressive Sensing

CS may be viewed as a dimension reduction technique that takes advantage of the fact that, in most cases, data with high frequency content is often not intrinsically high-dimensional. There is some kind of low dimensional structure in most signals which can then be exploited using low dimensional models. CS is a paradigm shift which looks at the possibility of operating at a low dimensional structure at all levels of the information processing pipeline from sensing, receiving of signals, to processing of signals [68]. In CS, data is directly sensed in a compressed form through the use of adaptive nonlinear projections which preserves the structure of the original signals. Thus, the original samples may be replaced with a more general form of linear measurements $y = \phi x$ which maintains all the relevant information content of x, where ϕ is an $M \times N$ measurement matrix, x is $N \times 1$ matrix of the original signals and $M \ll N$. A core theoretical question is we can design ϕ so that M is a small as possible. Tied to this is the question of how we can recover x from the measurements y. CS offers perfect reconstruction of the original signal provided certain conditions are met, and we briefly discuss below as we answer the above questions.

2.9.3.1 Sparsity

The concept of Distributed Compressed Sensing (DCS) is a relatively new field based on the notion of compressive sensing which states that if a signal is sparse, then its information content can be represented by a small set of linear projections which can, in turn, be used to reconstruct the original signal.

Consider a source generating some signal x and a collection of vectors $\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_n$. A signal χ, is said to be sparse if it can be approximated well by a few vectors from Ψ. According to CS theory, χ is K sparse if $\chi \approx \sum_{i=1}^{K} \theta_{n_i} \Psi_{n_i}$, where

$\| \theta \|_0$=K are the non zero entries, and K is small. It has been shown that if ϕ is $cK \times N$ matrix with iid Gaussian entries, then $c \approx log_2(1 + {}^{N}/_{K})$ would be an over sampling [69]. In my opinion, this approach is the most plausible in that even if there is redundancy due to either spatial or temporal correlations, then according to CS theory only information that is relevant to the reconstruction is extracted during sensing in the domain Ψ, leading to small energy expenditure during sensing, processing, transmission and reception.

2.9.3.2 Restricted Isometry Property (RIP)

The design of ϕ should satisfy what is known as the Restricted Isometry Property (RIP) [70]. If $\chi(t)$is a bandlimited signal, then

$$y[m] = < \phi_m(t), x(t) >$$

$$=\sum_{n=-\infty}^{\infty} x[n] < \phi_m(t), sinc\left(\frac{t}{T_s - n}\right) >$$

$$1 - \delta \leq \frac{\| \phi x_1 - \phi x_2 \|_2^2}{\| x_1 - x_2 \|_2^2} \leq 1 + \delta \qquad (27)$$

where $\| x_1 \|_0$, $\| x_2 \|_0 \leq K$.

If we start up with a pair of vectors that are K sparse and apply the matrix ϕ, then according to RIP even though we are mapping this set of signals to a much lower dimensional space, the distance between them does not change much. In other words, ϕ preserves the distance between them. The property is interesting in that after taking our measurements y, we may quantize them, add some noise somehow, but we do not want a situation where distant points end up being put together. If this were to occur, the implication would be that small measurement errors may lead to very large reconstruction errors.

2.9.4 Signal Reconstruction Methods

Most naturally occurring signals in science, Engineering and medical images exhibit sparsity and are hence compressible. The recovery of these sparse signals consists of solving NP-hard minimization problems. The methods for recovery of these sparse signals consists of Convex relaxations, Iterative Thresholding Algorithms, Greedy Iterative Algorithm, Combinatorial/sublinear Algorithms, Non Convex Minimisation Algorithms and F Bregman Iterative Algorithms. We briefly discuss some of these algorithms in the next subsections.

2.9.4.1 Convex Relaxation

Traditional CS signal recovery is achieved through expressing the CS problem as an L0-norm optimization problem with single sparsity constraints.

$$\min_{x} \| \Psi_x \|_0 \quad \text{s.t. } y = \phi x \tag{28}$$

Unfortunately, the problem in this current formulation is non-convex Non-Polynomial hard (NPH) problem. It can be alternatively relaxed to and expressed as an L1 optimisation problem as follows [71]:

$$\min_{x} \| \Psi_x \|_1 \quad \text{s.t. } \| y - \phi x \|_2 \leq \varepsilon \tag{29}$$

The algorithms achieve signal reconstruction by formulating the CS problem as a convex linear programming problem. Solving the L1 problem using linear programming is feasible but it is sometimes regarded as time consuming, thus the L1 problem can be transformed into more convenient formulations such as the L1 penalty formulation:

$$min_x = \| x - \emptyset x \|_2^2 + \lambda \| x \|_1 . \tag{30}$$

Other such typical examples of such approaches are the Basis Pursuit (BP), Basis Pursuit de-nosing (BPDN), modified Basis Pursuit de-nosing, Least absolute shrinkage and selection operator (LASSO) and the least angle regression (LARS). The basis pursuit decomposes the L1 norm $\| x \|_1 = u + v$, into two components, such that $u = \max(0, x)$ is the positive part of x and $v = \min(0,1)$ is the negative one.

BP and variants of BP manipulate interior point methods and simplex techniques to acquire optimal solutions.

The drawback behind these approaches is that although they require few measurements for reconstruction, they tend to be to computationally intensive. LASSO and LARS make use of pivoting operations on sub matrices to acquire solution paths to the convex optimization problem. These techniques are reported to be very efficient, especially in cases where there are high levels of sparsity.

2.9.4.2 Greedy Iterative Algorithms

This class of algorithms solves the CS reconstruction problem by making locally optimal choices at each step of the reconstruction process. The general idea is to choose a column of ϕ that has the highest correlation with **Y** at each iteration. This also includes the computation of minimum least square errors in each iteration. The chosen column's contribution from Y is excluded and further iterations are done on the remaining ones until a correct set of columns is identified. Typical greedy algorithms include compressive matching sampling pursuits (CoSaMP), orthogonal multiple matching pursuit. Although the stopping criteria may differ from one algorithm to another, these iterations are usually done for **M** iterations. In general, the benefit of greedy algorithms is that they tend to have low implementation cost and high speed recovery though signal recovery may be costly if the signal is not very sparse [72].

2.9.4.3 Iterative Thresholding Algorithms

Iterative thresholding algorithms is another approach to CS with algorithms that are generally faster than convex optimization problems. They solve the L1 penalized least squares problem in phases. Initially it optimizes the least square term by solving the optimization problem with L1 penalty, then the other phase is accomplished though threshholding the magnitudes of x entries. Some of the iterative algorithms in literature are iterative hard shrinkage, iterative splitting and thresholding, and message passing. The limitation behind these approaches is that in most situations it is difficult to demonstrate their convergence to a global optimum [73].

2.9.4.4 Bregman Iterative Algorithms

The Bregman iterative algorithms provide fast and proficient methods for solving the L1 minimisation problem. They are capable of providing exact solutions to the constrained L1 problem. They use a Bregman regularisation scheme which partitions the problem into sub-problems, which are then iteratively solved in a sequential manner to generate exact solutions to the constrained problem. The approach is based on the Bregman distance defined as follows [74]:

$$D_j^p(u,v) = J(u) - J(v) - < u - v, p >, \qquad (31)$$

where J is a convex function , $p \in \partial J(v)$ is a sub-gradient in the differential of J at the point v. $D_j^p(u,v)$ is a measure of the proximity of u and v since $D_j^p(u,v) \geq 0$, whenever $D_j^p(u,v) \geq D_j^p(w,v)$ for all points w between u and v but it is not an actual distance since $D_j^p(u,v) \neq D_j^p(v,u)$.

DCS that enables distributed algorithms to exploit multi-signal ensembles that exploit intra and inter signal correlations structure are reported in [75]

2.9.5 Comparison of CS with DSC

Traditional signal processing techniques such as DSC discussed early on rely on the application of Shannon sampling theory which states that a band limited time varying signal whose highest frequency is n can be perfectly reconstructed if it is sampled at a minimum time interval of $^1/_{2n}$ seconds. In this approach, "m" samples may then be used to compress, transmit and reconstruct the original signal, implying that the remaining "n-m" samples are discarded. This sharply contrasts with CS where just a few samples are used instead.

CS theory offers a better alternative to DCS in that we can compress and reconstruct the original signal from relatively fewer samples than that required by Nyquist theory and hence DSC is preferred. Moreover, the DSC approach apportions more computational work on the encoder so that the decoder can work in a more efficient manner. This completely contradicts the requirement of WSNs applications where the source side uses batteries that are power limited and therefore cannot manage large workloads. CS offers a better alternative since less work is done on the sensing side.

2.10 Data Aggregation Techniques

Data aggregation techniques exploit data redundancy exhibited by sensor nodes by selecting a subset of sensors to be active, thereby sensing transmitting and receiving data while other nodes are put to sleep to conserve energy. The active nodes are chosen in such a way that they sufficiently cover the whole network being monitored. The rationale behind this approach is that choosing only a sub set of nodes to be active suppresses redundant data from being collected thereby saving energy that would have been consumed during sensing and transmission or reception.

Recently, Paolo et.al., [76] presented a Tacit consent approach that aims to eliminate redundant transmissions from spatially correlated nodes in WSNs. In this approach the sensor nodes are divided into two groups, representative node and member nodes. Representative nodes are responsible for direct transmission of data while member nodes use overhearing to determine if additional information is required. Member nodes consent from transmitting their data if the estimation function correctly predicts their measurements. Although this approach is reported to have shown drastic energy consumption reduction, it may not be very suitable for applications where high precision of data is required due to its use of the estimation function. Various clustering algorithms also fall under this approach.

In an energy-aware spatial correlation, clustering protocol is presented in which cluster heads are responsible for portioning cluster members into correlation regions in such a manner that only one representative member of the correlation region can transmit information [77]. These representative members are chosen based on their residual energy and on a utility measure that aims to minimize expected distortion tolerance. Experimental results showed exponential decaying achieved distortion as a function of the degree of correlations, Figure 2.20. It was also established that energy consumption decreased exponentially with increase in desired distortion rates.

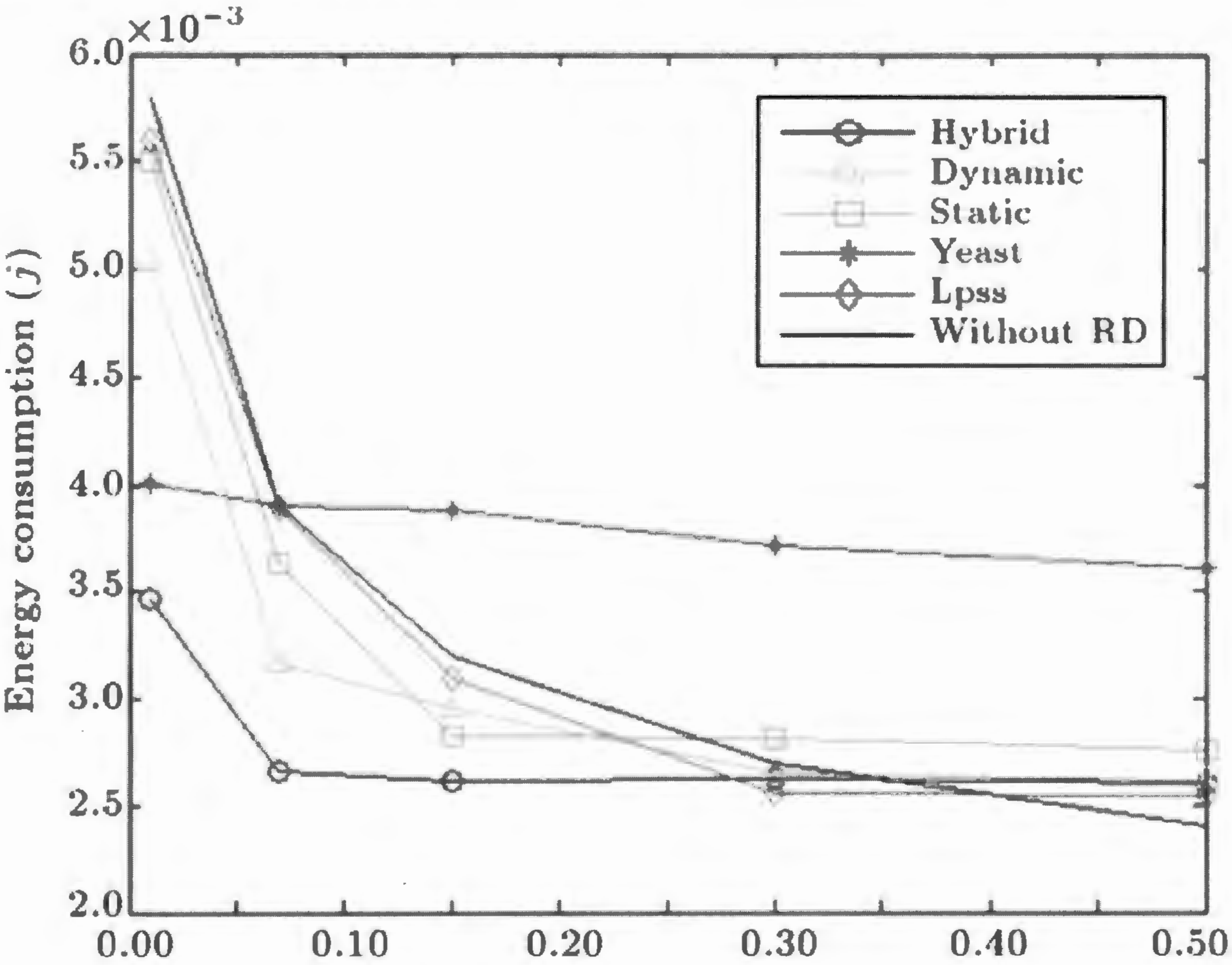


Figure 2.20: Energy Consumption with Observation distortions

Research efforts focused on energy-efficient data gathering in order to achieve desired distortion levels are constrained by the need for optimal coordination between routing and compression algorithms.

2.11 Temporal Correlations Models

Sensor Observation in event tracking applications where there is need to regularly sense the environment and send consecutive observations to the sink node are generally temporally correlated in nature.

Experimental results reported showed that the reduction of node density from 50 to 15 did not lead to any significant decrease in levels of distortion. There was only a steady and relatively constant reduction instead. This behavior was attributed to the fact that in densely populated network scenarios, the sensor nodes sent highly correlated redundant data which does not lead to any value addition in terms of resolutions.

2.12 Fast Fourier Transform and Distribution Processing

The computation of Discrete Fourier Transforms (DFT) is a significant component in many digital signal processing systems. Application of the fast Fourier transform (FFT) is increasingly gaining ground due to its ability to reduce the number of computations when computing discrete Fourier transforms. The widespread use of FFT has led to a variety of implementations, including custom FFT processors, digital signal processors, general purpose software and programmable hardware [78]. With the advent of Silicon on Chip (Soc) technology, DSP algorithms such as FFT are increasingly becoming popular and are used in processor cores which are embedded within the Soc Platform. The increase in demand of high performance, low power applications like multicarrier systems has led to an increase in demand for FFT/IFFT cores, which provide high throughput while minimizing power consumption. It has been shown that the main source of power consumption in a typical CMOS circuit is due to switching power, P_{sw} by:

$$P_{sw}=\frac{1}{2}kC_{load}V^{2}_{dd}f \qquad (32)$$

where V_{dd} is the supply voltage, f is the clock frequency, C_{load} is load capacitance of the gate and k is switching activity factor, which is defined as the number of times the gate makes an active transition in a clock cycle [79].

Genetic algorithms play an important role in the design and implementation of discrete time signal processing algorithms and systems. In [80], a genetic algorithm optimization for coefficient of FFT processor is reported. In [81], a multi-objective genetic algorithm for on-line adaptation of a multi-carrier code deviation multiple access (MC–CDMA) receiver is presented. The approach uses a genetic algorithm to adapt the receiver, while dynamically optimizing the Fast Fourier Transform section of the receiver for both error value and power consumption.

Figure 2.21 shows the block diagram of the Multi-Carrier based Telephone Receiver. It consists of an FFT block that is used to demodulate the orthogonal frequency division multiple access (OFDMA) and a combiner block which equalizes signals and separates the code users. The FFT processor is the most power consuming block, and its power consumption depends on the word length of data and the FFT coefficients. In [82], a Multi-Objective Algorithm is used to find the optimal word length for input data and Fast Fourier Transforms coefficients while satisfying functionality constraints. The limitation of this approach is that the fitness function is based on one objective function derived from some error term.

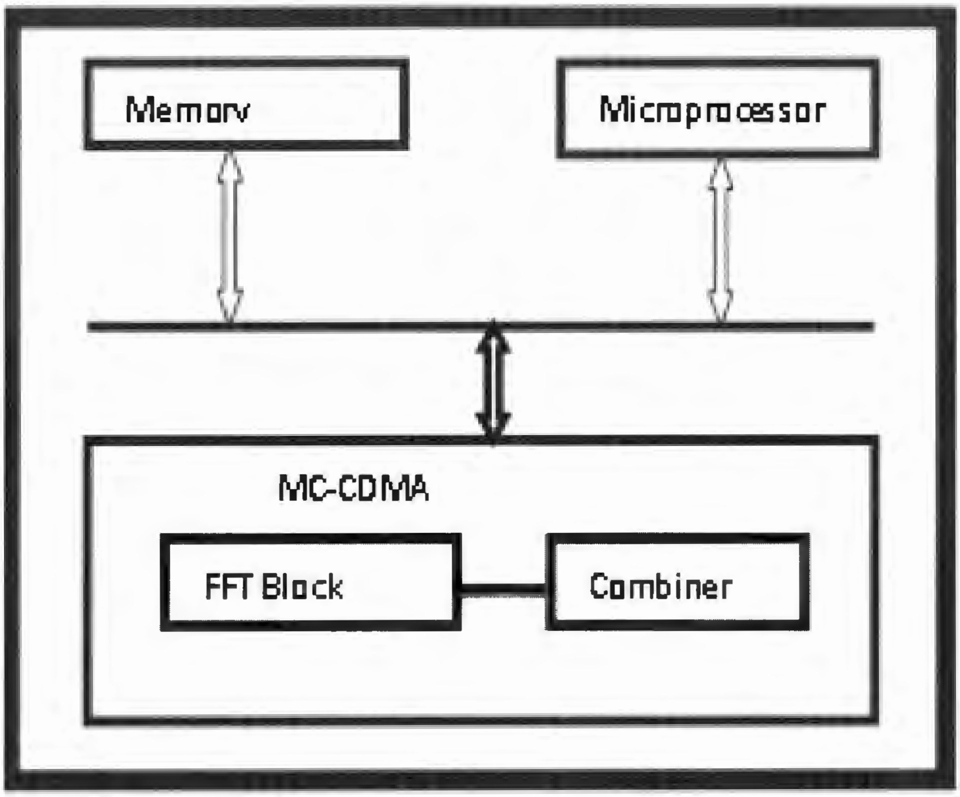


Figure 2.21: Multi-carrier code deviation multiple access telecommunications receiver

Processors with larger word length provide better performance in terms of accuracy. The accuracy of FFT processor is represented by Signal to Noise ratio (SNR), and a processor design with higher SNR value is desirable. On the other hand, a high word length in FFT design contributes to higher switching activities (SA). To resolve this problem, in [83] a single objective and a multi objective Genetic algorithm Approach is proposed to search for better SNR and SA value of the FFT processor.

2.13 Genetic algorithms in computing FFT

In [84], the idea of a Distributed Digital Signal Processor (DDSP is considered, where a network is composed of multiple sensor nodes, each of which corresponds to a processor. The idea of divide-and-conquer approach is then applied, in which a DFT of size N, a composite number is reduced to the computation of smaller DFTs from which the larger DFT is computed.

In [85], an Energy efficient algorithm for computing 1-D Fast Fourier Transform (FFT) over single and multi-hop wireless sensor network is proposed. The proposed algorithms reportedly reduces the number of transmissions, eliminates typical redundant computations in a distributed FFT algorithm and uniformly maps complex multiplications over all sensor nodes by introducing an extra bit-complement permutation stage after $(log_2 N)/2$ iterations.

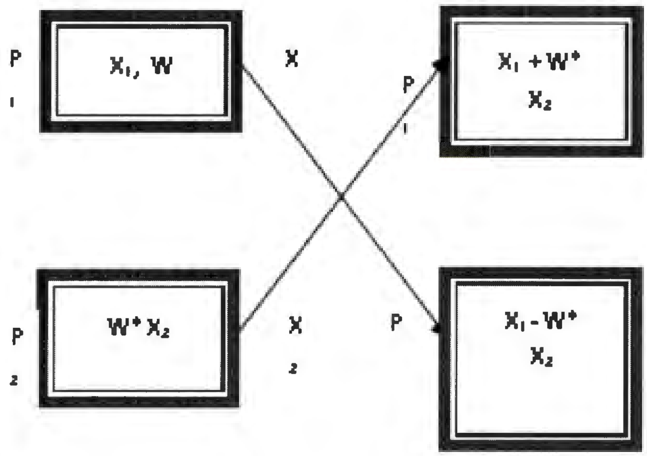


Figure 2.22: load balanced sensor network

Figure 2.22 above shows two sensors in a traditional load balanced network. An assumption is made that every sensor has a complex weight, w, required for the FFT for local computation. Between two consecutive stages, two sensors p1, and p2 have partial results, X1 and X2 available from the previous stage and they process information as shown in the diagram. The computations are as follows:

a) Sensor p_1 sends a copy X_1 to p_2

b) Sensor p_1 computes X_1=X_1+w*X_2

c) Sensor p_2 computes X_2 = X_1-w*X_2

It has been observed that at every stage, each sensor is involved in transmitting and receiving one packet over the radio channel, and a complex multiplication and one addition and multiplication. Note that this leads to a load balanced distribution of work, but with redundancy of computation w^*X_2 which is performed twice.

It has been shown in literature that transmission of a complex number over to an immediate neighbor consumes about 50 times more energy than the addition operation. Thus in order to reduce energy consumption, redundancy in complex multiplication should be avoided. There is also need to balance energy depletion among sensors. In an attempt to avoid redundancy in computation, the author in [81] proposed an approach that avoids redundancy at the expense of load balancing.

Figure 2.23 below shows a diagrammatic representation of the approach.

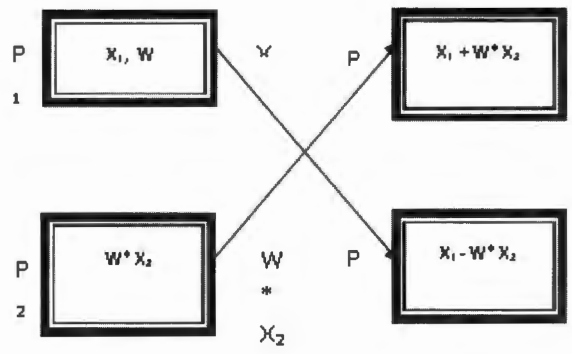


Figure 2.23 Two Sensors without redundancy.

In this approach, p_1 and p_2 are computed as follows:

a) p_2 computes w^*X_2

b) p_1 sends a copy of X_1 to p_2

c) p_2 sends a copy of w^*X_2 to p_1

d) p_1 computes $X_1 = X1+ w^*X_2$

e) p_2 computes $X_2 = X1-w^*x_2$

In the above scenario, we have transmission of two packets and both sensors are involved in addition/subtraction but only one sensor is involved in complex multiplication. This approach removes the redundancy of complex multiplications compared to the conventional load balanced algorithms. The limitation of this approach is its lack of load balancing; hence some sensors will deplete their energy before others.

The load balancing problem is resolved by proposing an algorithm in which sensors are labeled $p_1, p_2, \ldots, p_n$. Computation of data is done in one stage of the unbalanced algorithm, with even labeled sensors computing the complex multiplication and the odd labeled sensors using the results passed to them. In the next stage, the roles are interchanged and the odd labeled sensors compute the complex multiplication while the even labeled ones use transmitted results of the computation. The approach helps to achieve both load balancing and reducing redundancy of complex multiplication. The limitation of this approach is the cost associated with switching as roles are interchanged at the end of each stage. This may lead to larger memory requirements.

2.14 Low Power FFT Architecture

The evolution in semiconductor technology and the advert of SoC(System-on-Chip), has enabled prototyping of FFT algorithms as parameterisable cores which can be embedded within a SoC platform. There are common architectures for FFT processors: pipelined architecture and memory based architecture. Pipelined FFT processors provide higher performance and consume much more hardware resources than memory based FFT processors. In contrast, memory based processors need less hardware resources but require to operate at a higher clock frequency to meet throughput requirements.

In applications that require high throughput and low power consumption, the Pipelined FFT processor is the preferred core. However, in real-time applications, where data is a sequential stream, the pipelined architecture does not match the requirements of the FFT algorithms as it requires temporal re-ordering of data [86]. There is, therefore, need for a commutator, which is used to reorder the input data.

2.14.1 Processing Elements

As discussed in section 1.2, the basic processing unit of an FFT operation is a butterfly or processing element (PE). At the algorithm level, the first design issue considered is the selection of the radix of the FFT, which determines the time taken and area computation. Most researchers prefer the Radix- 4 due to its computational advantage over Radix-2.

One of the commonly used pipelined architectures is the Radix-4 Single path Delay commutator (R4SDC). This architecture is widely used due to its high utilization of multipliers, butterfly elements and memory blocks. While dedicated hardware can provide the highest processing performance, it suffers the drawback of inflexibility to changing processing requirements. While alterative systems based on software are flexible, they often suffer from insufficient processing capability. The radix operation evolves the use of look up tables (ROM) for complex multipliers, a technique used for twiddle factor multiplication in short length FFT. However, as the table size grows linearly for larger sizes of FFT, the efficiency of this approach decreases. Coordinate rotation digital computer (CORDIC) is an alternative technique to the approach discussed above.

2.14.2 CORDIC unit

CORDIC transformations offer one way to perform complex mathematical operations, such as division, square root, and trigonometric conversions, Figure 2.24.

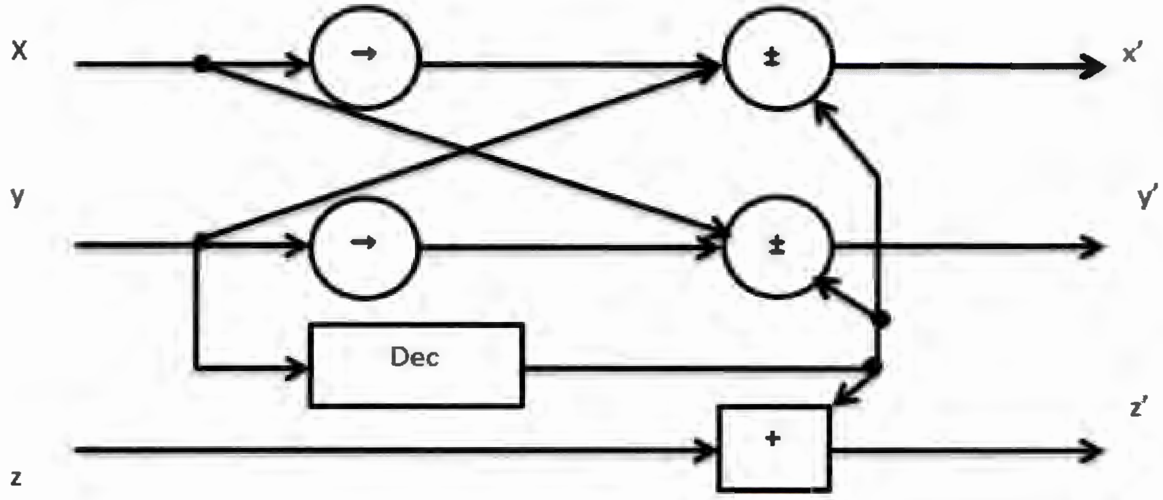


Figure 2.24: CORDIC Architecture

The basic CORDIC rotation in the form of a matrix is given by the equation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{23}$$

(x, y) and (x', y') are initial and final vector coordinates respectively and θ is the angle by which the vector is to be rotated and can be expressed in terms of the elementary angle α the equation

$$\theta = \sum_{i=0}^{b-1} d_i \propto_i \tag{24}$$

$\alpha_i = \tan^{-1}(2^{-i})$, $d_{i=sign(z_i)}$ and b is bit precision.

$$x_{i+l} = x_i - d_i 2^{-i} y_i$$

$$y_{i+l} = y_i + d_i 2^{-i} x_i$$

$$z_{i+i} = z_i - d_i \alpha_i \tag{25}$$

The CORDIC has been successfully applied in implementing FFT transform processors and other DSP applications as depicted in Figure 2.25. It replaces the sine and cosine twiddle factors of the conventional FFT implementations with CORDIC rotations. The Use of CORDIC in FFT algorithms utilizes only add and shift operators, thereby eliminating the need for complex twiddle factors and multipliers and hence saving power and space in Read Only Memory (ROM).

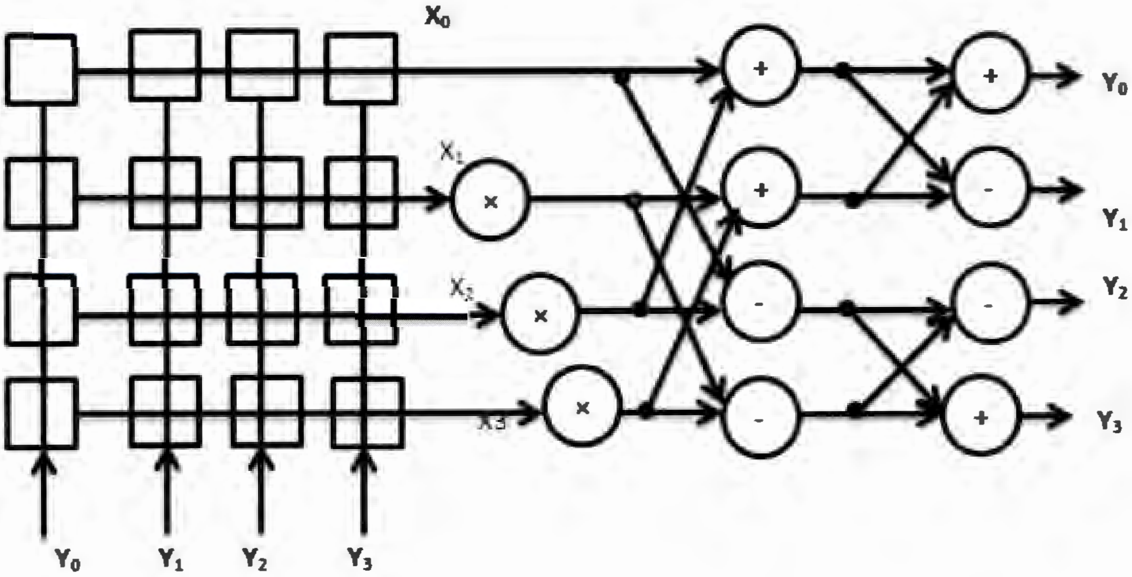


Figure 2.25: FFT Diagram

2.15 Evolvable Hardware and Field Programmable Gate Arrays

Computing research in the past decade has focused on dedicated processors and processor architectures. The computing systems designed strove to mitigate tradeoffs between flexibility and performance. At one extreme end of the design paradigm are Instruction-set architectures such as General Purpose Processors (GPS) and Digital Signal Processors (DSP). While this architectural design approach allows for arbitrary computations, its major drawback is inefficient performance and high power consumption.

On the other extreme end of the spectrum are application-specific integrated circuits (ASICs) which are optimally designed for specific tasks. ASICs are designed to be power efficient and have high efficient performance but they lack flexibility and are not reconfigurable. Reconfigurable computing has consequently emerged as an alternative to the two approaches as it lies in between the two design paradigms.

Thus, while research in the past decade has mainly focused on dedicated processors and ASICs, a new approach called Evolvable hardware (EH) has emerged. It is a new concept where the structure of hardware is designed to adapt to changes in task requirements or changes in the environment through its ability to reconfigure its own hardware structure online and autonomously too.

The application of evolvable algorithms started in the 1990s when researchers used them to design computer chips that can dynamically change their hardware functionality. This combination of programming electronics such as Field programmable gate arrays and programmable logic circuits lead to a new field now called evolvable hardware. EH may be defined as the design or application of Evolvable applications and biologically inspired algorithms for the purposes of creating physical devices or optimized physical devices. When designing EH applications, it is important that designers ensure that they do not just achieve biological inspiration but we also benefit from the evolved hardware.

In [87] a combination of genetic algorithms and software reconfigurable devices are used to design virtual reconfigurable hardware. The structure of reconfigurable devices is determined by downloading bit strings called architecture bits. The main drawback of this approach is its reliance on downloading and uploading of bit streams online. Hybrid reconfigurable systems that comprise general purpose processors combined with FPGAs have achieved tremendous success in accelerating computational performance of performance-critical applications.

Considerable effort has been made in exploring parallelism in the design of efficient hybrid reconfigurable systems, ranging from multiple parallel granularities to fine grained parallelism inherent in FPGAs, to course grained parallelism as in multiple-interconnected CPU-FPGA nodes.

The design of CPU-FPGA architectures is limited by the drawback associated with inefficiencies in data transfer between CPUs and FPGAs components. The computation of an FFT by FPGA is motivated by the ease of achieving high speed computation due to the inherent parallelism in FPGAs. Reconfigurable computing systems vary in complexity depending on their functionality, application requirements and computational approach but the critical components could be categorized as compilers that enable easy and fast compilation of high-level software descriptions into hardware circuits and run-time reconfigurable mechanisms, usually availed through CAD tools. Figure 5 shows an example of the architecture of a reconfigurable system.

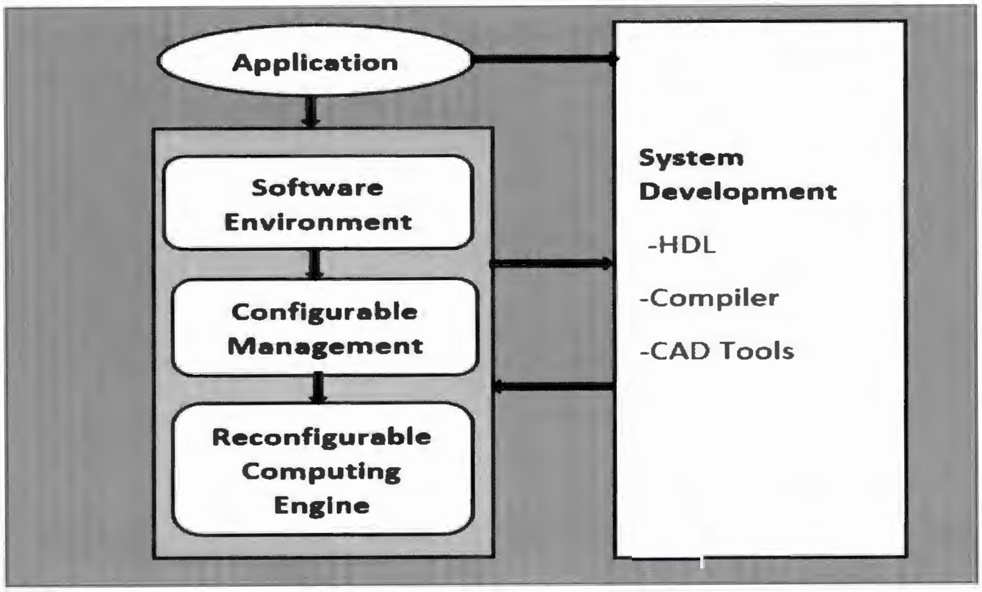


Figure 2.26: Architecture of a Reconfigurable System

2.16 Field Programmable Gate Arrays.

Per Vanderbauwhede [88], Field FPGAs were pioneered by Hilinx in the early 1990s. They comprise of a rectangular micro-chip which is surrounded by a ring of input/output blocks on its outer edge. The IOB facilitates access to a variety of selectable input\output pins located in the interior of the FPGA. A typical logic block may comprise of input lookup tables, flip-flops and other primitive logic gates. Modern FPGA designs comprise high performance DSP blocks, embedded memories and thousands of other configurable logic blocks (CLB) and programmable interconnectors.

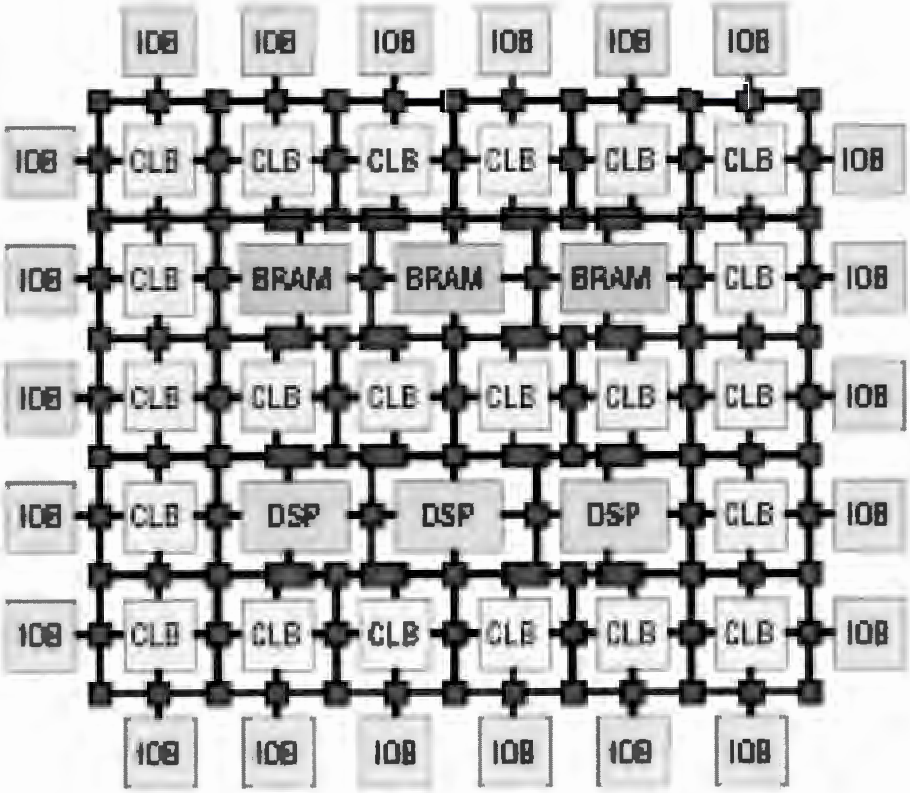


Figure 2.27 FPGA Structure

2.17 The need for context-awareness in WSNs

In a wireless sensor network, sensors are deployed within the area of interest subject to observation and these are left with no monitoring. These sensors are normally battery powered, and they are always, if not in most cases, transmitting data from the area of interest to the base stations for processing. These transmissions put more stress on the battery of the sensors; hence the problem of energy constraints becomes a major concern in terms of limited battery lifetime. One of the major concerns in the design and deployment of wireless sensor networks is directed at maximizing network life time. Network life span is decided by network connectivity and network connectivity in turn depends on network bottleneck nodes [89].

In multi-hop wireless sensor networks, nodes that are close to sink node transmit much more data, and then exhaust their energy while other nodes in the same network remain without energy. It is therefore necessary to research on the various ways of predicting the energy consumption in a wireless sensor network to enable the adoption of appropriate distribution strategies that extend the lifespan of a network as much as possible [90]. Literature reveals that in most of the existing protocols on WSNs, context-awareness is exploited in a single dimension and it is implemented either at the application layer or at the routing layer using a single context parameter.

The concept of context aware computing, in which WSNs nodes may be deployed and customized to use their changing contextual information such as residual power, channel quality, spatial location, sensor roles and sensor mobility to reconfigure themselves and adapt their behaviors (eg sleep modes, transmission\receive, sampling rate etc.) is an increasingly important area of research.

Bill Schilit [90] introduces the concept of context–aware computing as follows: "Such context-aware software adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to things over time." Anind Dey [91] describes context as: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." We argue that good WSNs should have capabilities to examine their computing environment and react to the changing environment. In this research, we argue that context is not only limited to location of sensor devices, but other parameters such as residual power, channel quality etc.

We argue that if a sensor device is aware of its computing environment, then it can use this information to adapt its behavior by providing services in a way that the network life time could be prolonged.

The creation and deployment of complete systems to evaluate the energy consumption of wireless sensor is costly, time consuming, and it requires substantial domain expertise. Naïve approaches in which nodes periodically send information about residual energy to a monitoring node are impractical as so much energy is spent due to communications that probably the utility of residual energy information would not compensate the amount spent in the process [92]. Due to these challenges, it is desirable to capture properties and behavior of particular aspects of the network systems through model abstractions and frameworks.

Markov models have long been used to model exponentially distributed systems with arrival times and services times. WSNs may be viewed as exhibiting such characteristics. Markov models are composed of event chains where the likelihood of a system being in a particular state is independent of it being in any other state. This restriction is a very big drawback to the application of Markov models in WSNs [93]. In this research, we argue that the probability of a sensor node to transmit from one state to another should depends on the context in which the node is operating under.

2.18 Context-Aware Modelling

Dynamic feature and large scale deployment of WSNs require sensor nodes to adapt capabilities of self-organization, self-configuration and self –coordination. Context-aware computing is the ability of applications to discover and react to changes in their operation environment. In this paper we integrate the concept of context aware computing with WSNs, and theory of stochastic processes to build a stochastic context Aware Energy Consumption and distribution Model for Multi-hop WSNs. Context awareness is one of the methods used to improve resource allocation in Distributed Sensor Networks (DSNs).

In [94] a mobility-aware protocol for WSNs in Health –care monitoring is proposed. The proposed protocol has mechanisms for addressing common sensor behavior such as data exchange in the context of sensor mobility. The mechanism minimizes sensor collisions and optimizes sensor duty-cycle of each node. The reported results shows limited reductions in energy consumption compared to other protocols. When designing context-aware applications, it's important to extract the different circumstances under which the application is running, processes them and deduces what should be performed in a particular situation. While most of the contextual logic may be specified by the application user, some may change over the lifespan of the application, hence the need for context mining and discovery.

2.19 Overview of Stochastic Modelling Processing

Markov models have long been used to model exponentially distributed systems with arrival times and services times. WSNs may be viewed as exhibiting such characteristics. Markov models are composed of event chains where the likelihood of a system being in a particular state is independent of being in any other state. This restriction is a very big drawback to the application of Markov models in WSNs. In this paper, we argue that the probability of a sensor Node to transit from one state to another should be context dependent unlike in Markov Models. Context based simulation models may provide better energy conservation results as systems reach steady state probabilities of the system.

In [95], a probabilistic approach to predict the energy consumption in WSN is presented. The approach uses a Network State Model to model the behavior of sensor nodes and then predict energy consumption by a sensor node. The rationale behind this approach is that in many instances the node can predict its energy dissipation rate based on its own past history and also on the past history of its neighbors. If a sensor can efficiently predict the amount of energy it will dissipate in future, then it won't be necessary to transmit available energy often. The node will simply send its available energy and the dissipation rate.

2.20 Analysis of stochastic Modeling

A Markov process is a stochastic process (random process) in which the probability distribution of the current state is conditionally independent of the path of past states, a characteristic called the Markov property. A *stochastic process* in a discrete time $n \in \mathbb{N} = \{0,1,2,...\}$ is a sequence of random variables X_0, X_1, X_2, ...denoted by $\boldsymbol{X} = (X_n: n \geq 0)$. We refer to the value X_n as the *state* of the process at time *n* with X_0 denoting the initial state. Stochastic processes are meant to model the evolution over time of real phenomena for which randomness is inherent [96]. The behavior of sensor nodes as it is receiving and transmitting data may have been observed to follow Markov processes. A typical sensor node may have the following modes [97] :

Mode1: sensing on, radio transmitting;

Mode 2: sensing off, radio transmitting;

Mode 3: sensing on, radio receiving;

Mode 4: sensing off, radio receiving;

Mode 5: sensing on, radio off;

Mode6: sensing off, radio off;

A sensor node with M modes of operation can be modelled by a Markov chain with M states. To model such a network, for each node we denote a sequence of random variables, X_0, X_1, X_2, ...X_n that represents the states of this node in time. The probability that a node currently in state *i*, will next be in state *j* in the next transition is called the transition probability, and it's denoted by:

$$P_{ij} = P\{X_{m+1} = j | X_m = i\} \tag{33}$$

Similarly, a two–stage transition probability $P_{ij}^{(2)}$, that denotes that a node currently in state *i*, will next be in state j after two additional transitions is denoted by:

$$P_{ij}^{(2)} = P\{X_{m+2} = j | X_m = i\} \tag{34}$$

$P_{ij}^{(2)}$ can be computed from P_{ij} to give:

$$P_{ij}^{(2)} = \sum_{k=1}^{M} P_{ik} P_{kj} \tag{35}$$

The n-stage transition probability, otherwise known as the Chapman-Kolmogorov equations is defined as:

$$P_{ij}^{(n)} = \sum_{k=1}^{M} P_{ik}^{(r)} P_{kj}^{(n-r)} \tag{36}$$

For values of r such that $0 < r < n$.

The author in proposed a stochastic model of WSNs in which each sensor randomly and alternatively stays in an active mode or a sleep mode. The active phase is decomposed into two phases, the full-active phase and the semi-active phase as indicated in figure 1. When the sensor node is in full-active phase of the active mode, it may sense data packets, transmit the sensed packets, receive packets and relay received packets. When in semi-active phase, it is only able to transmit/relay data. When the sensor node is in sleep mode, it does not interact with the external world.

The duration of the sensor node in the sleep mode, full-active phase is exponentially distributed with mean $1/\beta$, and $1/\alpha$ respectively. In the full-active phase, a sensor node may generate or relay data packets according to a Poisson process with rate λ and λ_E respectively. In the full-active phase, a sensor node may also transmit/relay data according to an exponential process with mean $1/\mu$.

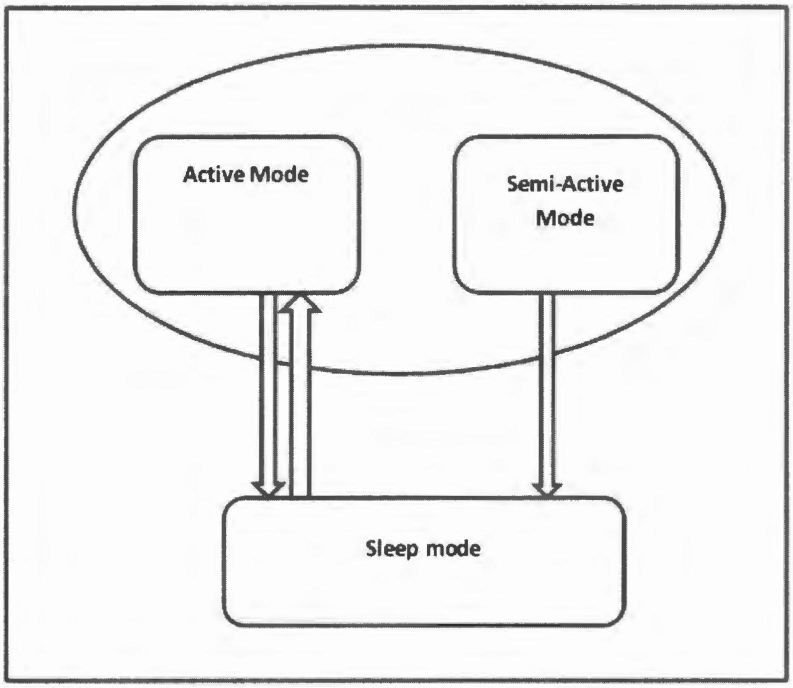


Figure 2.28 Transition Diagram for a Sensor Node

After some time in the full-active phase, a sensor node may move to sleep node if there are no data packets awaiting processing. If there is at least one data packet waiting processing, then the sensor nodes moves to semi-active mode where it can only process or relay data packets according to an exponential process with mean$1/\mu$, but it can't generate or receive any packets. While the approach by the author is plausible, there is need to evaluate other alternative distribution that can be used to modeling the duration of stays of the sensors in each of the phases as well as the processing times in order to identify the most appropriate model.

2.21 Analysis of Stochastic Energy Consumption

The probabilities of the Markov chain can be represented as a matrix P with M rows and M column. The author in argues that knowledge of these transition probabilities can then be used to represent the behavior of sensor nodes.

The expected amount of energy spent in the next T times, E^T can be computed as:

$$\boldsymbol{E^T} = \sum_{S=1}^{M}(\sum_{t=1}^{T} \boldsymbol{P_{is}^{(t)}}) * \boldsymbol{E_s} \qquad (37)$$

Where E_s is the amount of energy dissipated by a node that remains one time-step in state *s*, and $\boldsymbol{P_{is}^{(t)}}$ presents the probability that node currently in state *i* will be in state *s* after *t* transitions.

In [98], a comprehensive cross-layer analysis framework is presented in which the author employs stochastic queuing models to derive the distribution of energy consumption for nodes in a WSNs during some given time period. The energy consumption distribution was derived in [12], by consideration the amount of energy ε_i contained in a sensor node in a state *i* of X_n at the beginning of each time unit T^u. The probability distribution function *pdf* of the energy consumption during this time period is given by $g_i(e) = \delta(e - \varepsilon_i$ where $\delta()$ is the delta function.

Denote $h_{ij}^1 = g_i(e)P_{ij}$, where P_{ij} is the *(ij)*th element of the transition matrix (4). Then $h_{ij}^1(e)$ is the pdf of energy consumption given the Markov process. Then the joint conditional pdf becomes

$$h_{ij}^T(e) = \sum_{k\in S}(h_{ij}^1 * h_{ij}^{T-1})e \tag{38}$$

where S the set of all sensor states in X_n. Given that the life time distribution for a given node, **L(c),** is a function of its total energy **C**, it can be shown that the cumulative distribution (*cdf*) of the node's life time is:

$$F_{L(C)}(\mathrm{t}) = \Pr(E(t) > C = \int_C^\infty f_{E(t)}\,(e)de, \tag{39}$$

where $f_{E(t)}\,(e)$ is the pdf of energy consumption for any given period T. When t is sufficiently large, **E(t),** approaches a normal distribution with mean $\mu(t)$and variance $\sigma(t^2)$. Thus the cdf of lifetime can be approximated as:

$$F_{L(C)}(t) \approx Q(\frac{C-\mu(t)}{\sqrt{\sigma^2(t)}}). \tag{40}$$

In [14] an energy efficient node distribution model for WSNs is presented. The model assumes a monitoring area as a circle of radius r with sink node on its center. The approach derives a maximum no of hops as: $\boldsymbol{h_{max}} = \frac{\boldsymbol{r}}{\boldsymbol{d}}$, where r is the radius of the monitoring sensor area and d the hop distance. The approach then derives a telecommunications energy model based on the analysis of average energy consumption on sensor nodes deployed. According to Verdone etal [12], the average energy consumption percentage of an h-hop layer is:

$$\overline{\boldsymbol{E}} = \frac{E_h}{n_h} = \frac{mE_T}{n_h n E_{max}} (\sum_{i=h}^{h_{max}} \boldsymbol{n_i} - \frac{n_h}{2}) , \qquad (41)$$

Where n_i refers to particular nodes, n_h refers to number of nodes in hope layer h, E_T is energy consumption in a routing path.

2.22 Chapter Summary

This research reveals how genetic algorithms have been used to model sensor communication, in clustering and routing problems. We also provide a performance evaluation of various GA-based optimization strategies. Our observations show that while a number of algorithms try to select the best cluster headers or routing path based on some metric, the process normally introduces overheads in communication which in turn leads to more energy dissipation. We propose that future research should focus more on the use of_Stochastic Network State Model to model the behaviour of sensor nodes and then predict energy consumption by a sensor node with minimum overheads in communications to base station. Genetic algorithms apply when the elements are real discrete or complex valued. Thus genetic algorithms are suitable for digital signal processing and fast Fourier transform computations.

This chapter reviewed the problems encountered in optimisation of multiple conflicting objectives as applied to WSNs. The chapter reviewed how genetic algorithms have been exploited to establish energy efficient WSN protocols. This is accomplished by initially discussing the multi-objective optimisation problem in its general form and then later discussing how it has been applied to optimisation of WSNs algorithms. The chapter also discussed applicability of optimisation strategies in Quality of services routing protocols.

The chapter also reviewed various load balancing approaches used for WSNs, focusing on Tree based load balancing and models based on Channel state information.

This chapter evaluated the techniques that could be applied in WSNs in order to minimize the transmission of redundant data in order to minimize energy consumption. It mainly discussed techniques that can be exploited to reduce the transmission of spatially and temporally correlated data in WSNs.

The major efficient techniques discussed are distributed source coding which is an extension of the Slepian-Wolf coding technique and Compressive sensing. We evaluated the use of signal reconstruction methods such as Convex Relaxation, Greedy Iterative Algorithms, Iterative Thresholding Algorithms and Bregman Iterative Algorithms.

Our evaluation shows that Greedy Algorithms are the most suitable method to use since they have low implementation cost and high speedy recovery. However, their signal recovery may be very costly if the signal is not sparse.

Our evaluation shows that compressive sensing achieves better performance than distributed source coding and other competing techniques since it is capable of achieving good signal reconstruction with fewer samples than other techniques.

Chapter 3

Multi-Objective Genetic Algorithms and Field Programmable Gate Arrays

3.1 Chapter Overview

In this chapter, we present the Multi-Objective Genetic Algorithm used for computing FFTs using FPGAs. We initially present our gene encoding approach and the chromosome structure that represents a typical circuit used for computing FFT on FPGA in a non-optimal manner. Programmable logic units, (Luts-representing genes in a chromosome) were designed using schematic /block diagrams and hardware description language (VHDL) and they were encoded into chromosomes and then optimized using field programmable gate arrays (FPGA) to achieve desired goals of portability and low energy consumption. We then present the genetic algorithms used to evolve the FPGA circuits to compute the FFT optimally. Finally, we simulated our FPGA using Altera Quartus II 13.0.

3.2 Gene Encoding and Chromosome Structure

The Multi-Objective Genetic Algorithm Field Programmable Gate array design (MOGAFPGA), implementation starts with a pool of chromosomes which represent a feasible solution to the optimization problem. The implementation started with feasible chromosomes and then moved towards optimality. Each chromosome is made up of several genes that define the chromosome length. Figure 3.1 shows a typical chromosome and its constituent genes.

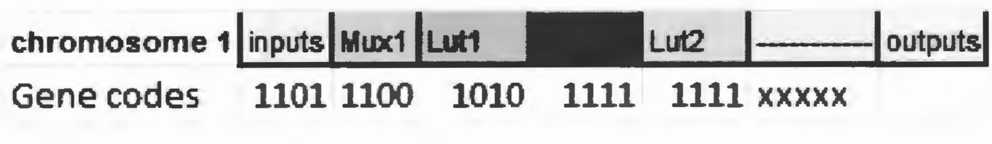


Figure 3.1: Chromosome with several genes (programmable logic units).

Each gene represents a programmable unit, which is a combination of two multiplexers and combination of a number of primitive logic units that are connected together to perform a specific function normally referred to as processing elements (PE). The manner in which genes are represented is very critical to the evolution and validity of the emulated circuitry. In our approach, we represent each logic unit as a triplet.

$\langle f_j | n_j^i | m_j^i \rangle$ for i=1,2 and 3 for each jth value, where $f_j$denotes the output from the jth processing element PE, n_j^i, denotes the ith input of the upper multiplexer of the jth PE, and m_j^i denotes the ith, input to the lower multiplexer of the jth PE. Our chromosome is represented as a sequence of such triplets as indicated in equations 42.

$$[\ f_j \quad (n_j^1, n_j^2\ , n_j^3) \quad (m_j^1, m_j^2, m_j^3)|, \ldots, |f_{j+N-1}(n_{j+N-1}^1 n_{j+N-1}^2\, n_{j+N-1}^3)\ (m_{j+N-1}^1 m_{j+N-1}^2, m_{j+N-1}^3)]$$

(41)

The chromosome forms the netlist input to the nios II simulator for fitness evaluation and selection.

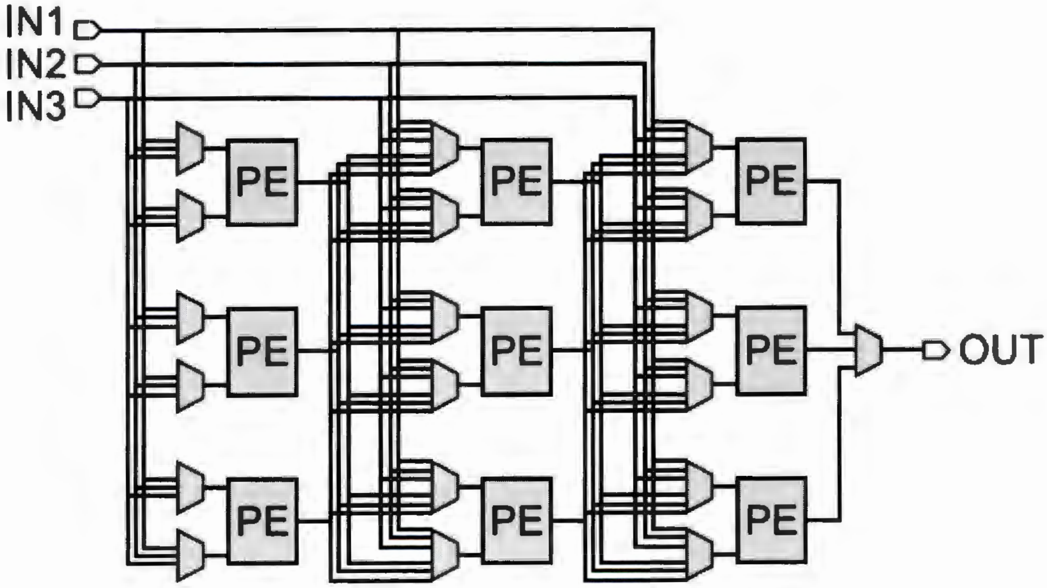


Figure 3.2: Used FPGA architecture

The Lut structures were incrementally built using Verilog VHDL and Quartus Schema/block diagrams interchangeably.

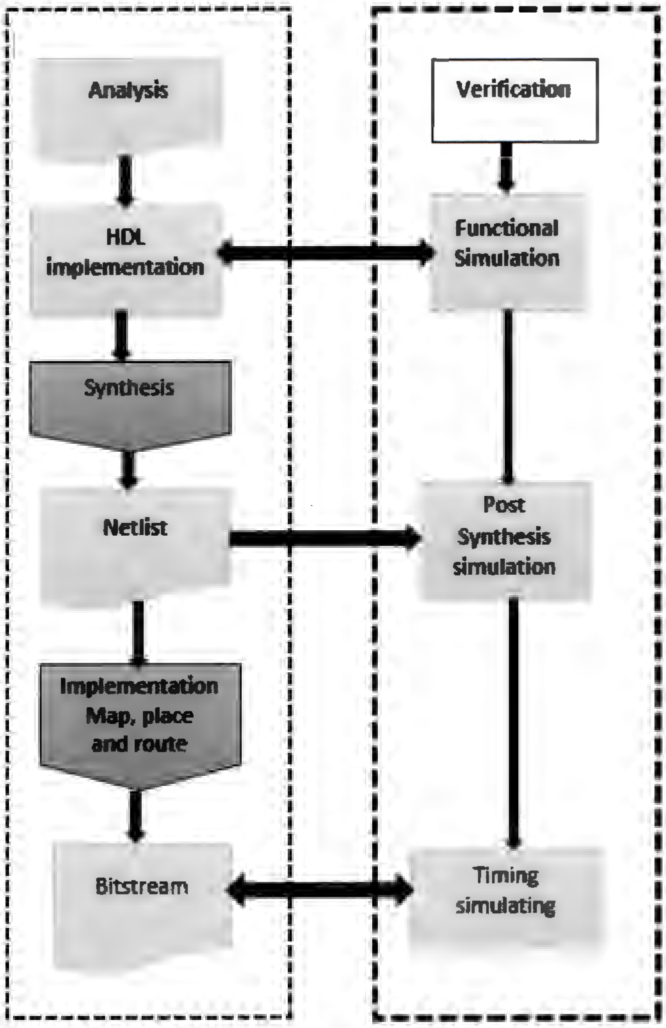


Figure 3.3: Simulation Diagram

The built-up of the Lut structures involved Compilation, Analysis and synthesis stages to verify the function correctness of the components. Figure 3.4 shows compilation results for a sample Lut.

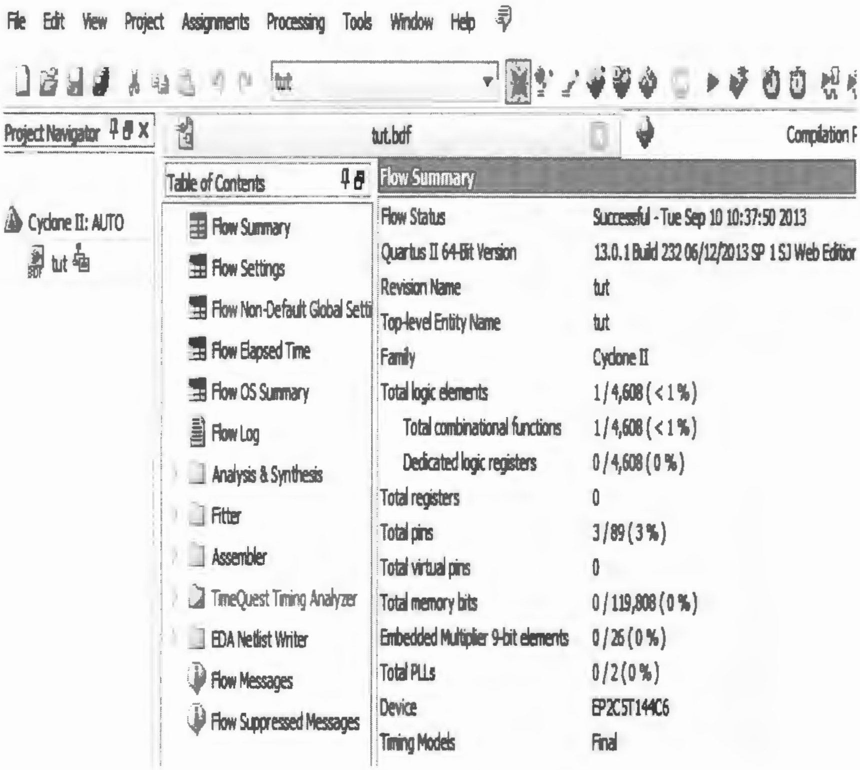


Figure 3.4 Compilation results for a sample Lut.

The Lut compilation above was performed using Quartus II 64-Bit version, device EP2C5T44C6. The compilation shows that there was no need for embedded multiplier elements, no need for dedicated registers and no need for virtual pins to process the Lut function. Only less than one percent of total logic elements were utilized using 3 pins. This compilation shows that the Lut structure under consideration is not resource wasteful and hence its inclusion within a reconfigurable circuit would be energy conservative.

The gene codes represent the configuration of each logic unit. The processes of mutation are used to change gene codes, which results in reconfiguration of a logic unit such that the Lut behaves differently. The process of crossover between two parent chromosomes is used to generate a child which represents a new solution to the problem. The processes of mutation are carried out until the number of child chromosomes reaches a specified number, upon which the process is stop and selection takes place.

3.3 Multi-Objective Genetic Algorithm based on Field Programmable Gate arrays.

The Multi-Objective Genetic algorithm based on Field Programmable gate arrays (MOGAFA) is implemented using three major components, namely the main algorithm which named Genetic Algorithm 1, the population selection and mutation algorithm which we named algorithms 2, and the crossover algorithm which we named algorithm 3. The main algorithm is as follows:

Genetic Algorithm 1

M: Maximum number of generations

P: Population size

N: Number of genes

Pr: Probability of cross over or cross over rate

Prm: Probability of mutation or mutation rate

```
Begin Algorithm
        initialise_Population ()
        for generation = 1 to N  do
            evaluate_population_fitness_function  f_n();
             reproduce population();
             for i=1 to P/2  do
             crossover(Pr);
       end for
      for j=1 to N
   mutate(Prm);
   End for
       for i=1 to POP_SIZE do
              Local_ improvement(Plocal);
              End for
              Elitism ();
  End for
End algorithm
```

We randomly select *n* chromosomes of Length N and fit them onto the FPGA using algorithm 2 (Population selection). We then use the FPGA to compute the FFT equation. We then evaluate fitness function

$$minimise\ f_n = \sum_{j=1}^{N}\sum_{i=1}^{N} \mu\, V_d^2 \ + w_{ij} l_{ij} f_j + x_{ij}\, y_{ij} P_j \qquad (42)$$

Where μ, is a constant denoting capacitance, V_d is voltage term relating to power consumption, f_j is the of logic slices on the entire FPGA space, relating to portability measure and w_{ij} , l_{ij} are the width and length in μm of the Luts used, and x_{ij} , y_{ij} are the length and width of the multiplexor used, P_j.

Algorithm 2 Population selection

Input: Integers $0 \leq n_{ij}^{1,2} \leq N + 1$

Output: P: Population; G: Generation

1: **for all** g such that $1 \leq g \leq G$ ***do***

2: $P_g = P_i + P_{mut}$

3: $f_g = \tau(P_g)$

4: $P_{ng} = \Lambda[P_g(\ldots P)]$ {sort and sect P}

5: **if** g =0.1G **then**

6: $P_g' = P_i + P_m$ {variation and mutation}

7: **end if**

8: **end for**

Algorithm 2 is used for population selection, mutation and sorting to produce a new population generation. The algorithms select a small sample of the population and mutates it to produce new chromosomes which are then added the larger pool of chromosomes. We then perform crossover with random probability *pr* using algorithms 3 (cross over algorithm).

Algorithm 3 Crossover

Input: Integers: $0 \leq n_{ij}^{1,2} \leq N + 1$

Ensure: Network connectivity and $n_{0j}^{1} = 0$.

for all i such that $0 \leq i \leq P$ **do**

  for all j such that $0 \leq j < \delta_j$ **do**

    $L'_{ij} = R(L_{ij} \ldots L_{ij} + N - 1)$

    $n_{ij} = \varepsilon\,(n_{ij}^{1,2,3})$

    $m_{ij} = \varepsilon\,(m_{ij}^{1,2,3})$

    if j=0 then

      return $n_{i,j}^{1} = 1, m_{i,j}^{1} = 1$

    end if

  end for

  for all k such that $\delta_j \leq k < N$ **do**

  $L'_{ij} = R(L_{ij} \ldots L_{ij+1} + N - 1)$

  $n_{ij}^{1,2,3} = R(n_{ij}^{1,2,3} \ldots n_{ij+\delta_j}^{1,2,3})$

  $n_{ij}^{2} = \varepsilon(n_{ij+1}^{2})$

  $m_{ij}^{1,2,3} = R(m_{ij}^{1,2,3} \ldots m_{ij+\delta_j}^{1,2,3})$

  $m_{ij}^{2} = \varepsilon(m_{ij+1}^{2})$

  end for

end for

To create the initial population, the first node, n_1^0, and m_1^0, for the ith chromosome to which the pMos and clock signal is connected is fixed to 1. This ensures continued connectivity to the source and drain of the pMos transistors. The second, third and subsequent nodes are randomly chosen from the chromosome positions between 2 and N.

3.4 Chapter Summary

The chapter presented a Multi-Objective Genetic algorithm for computing FFT. The chapter presented a gene encoding scheme for representing the various logic components required to compute FFT on FPGA. The Lut structures were incrementally built using Verilog VHDL and Quartus Schema/block diagrams interchangeably. Lut compilation were performed using Quartus II 64-Bit version, device EP2C5T44C6 to check the function correctness of the components. Based on the Lut structures built, chromosome structured for configuring and computing FFT on FPGA were constructed and evolved using selection, cross over and mutation of Lut structures on FPGA. The Multi-Objective genetic algorithm was simulated using Altera Quartus II 13.0.

Chapter 4

Stochastic and Context Aware Models

4.1 Chapter Overview

The chapter presents a Stochastic Context-aware Energy consumption and distribution model (SCAEDM) for WSNs. We also present an architecture for SCAEDM and its main components. We discuss how SCAEDM performs context scheduling, context processing, context adaptation and routing and media access among other processes. The chapter presents a SCAEDM algorithm for packet forwarding and routing based on Geographic Opportunistic routing and Energy Consumption Distribution function derived from multiple context based information such as link quality and residual energy.

The chapter also presents a SCAEDM context-aware energy consumption and duty scheduling algorithm which allows sensor nodes to a make optimized decisions by automatically adapting their protocol behaviour through considering runtime contextual state information. The chapter also presents algorithms for knowledge discovery and association rule mining designed to identify factors that influence the node behavior and hence energy consumption.

4.2 Stochastic Context-aware Energy Consumption and Distribution Model

SCAEDM is built upon the notion of discovering contextual information that describes the various situations a sensor node finds itself in and how it then uses this information to adjust its behavior in order to save energy and balance network load. Our model is motivated by the works of [10] and we build upon and extend this work. Our models address the following concerns:

- ❖ The ability of running application to determine lower layer decisions and hence control network behaviour.
- ❖ Integrating main context parameters of interest in WSNs such as residual energy, delay, mobility, spatial orientation, channel quality, Context discovery

and context communication (between nodes and inter-layer context sharing), load sharing and load balancing across networks.

4.3 Architecture of the Model

The SCEDM framework is built upon the notion of discovering contextual information describing the various situations that a sensor node or a sensor network finds its self in and using this information to select the most appropriate set of actions and/or behaviours to perform in order to save energy and hence prolong the network life time. As illustrated in Figure 4.1, SCAEDM has the following main components: context knowledge base, context collectors, context adaptation layer, context Engine, Applications layer and the Routing and Media access layer. Mining sensor association rules involve collecting the behavioural data which shows the sensor activities and finding association rules from the behavioral data.

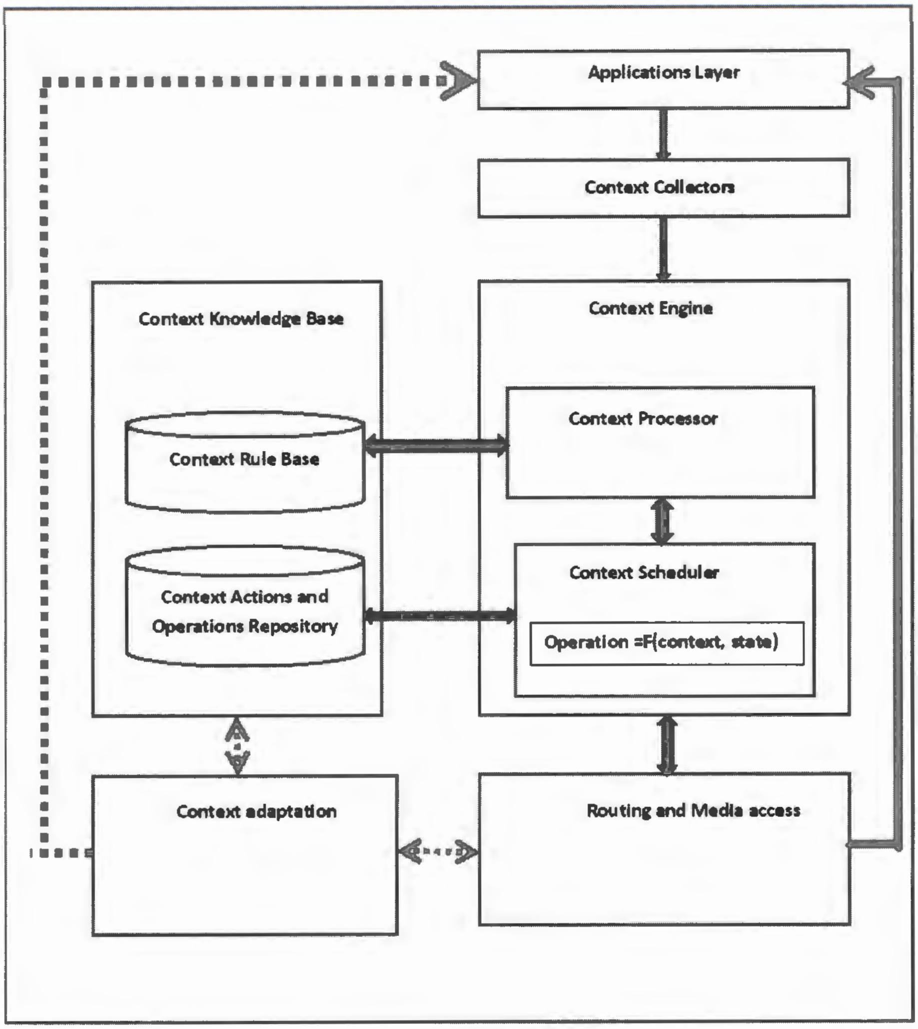


Figure 4.1 Context-aware Framework for WSNs

Context engine is composed of two components: the context processor and the context scheduler. The context processor uses association rule mining to mine context rules that govern how the sensor node should behave. Mining sensor association rules involve collecting the behavioral data from the Applications Layer, which shows the sensor activities, and finding association rules from the behavioral data. The knowledge base comprises two components, the context rule base and the context Actions and Operations Repository. The former is responsible for storing mined context rules while the latter stores corresponding context actions and operations derived from the context processor.

The mine rules are then passed to the context scheduler which is responsible for implementing the context actions and operations to be performed. This function resides at the cluster header which has higher energy level than others. It also has a standby backup node that receives updates on mine rules so that when it takes over the context scheduling processing it would have all the requisite information.

These rules are in the form of predicate logic/first order logic/ uncertainty reasoning eg *if residual energy (Node5,50), and sampling rate (node5, 40) then sleep_parameter_probability (Node5,0.2) and arriva_rate (Node5, probability (Poisson, 0.3))*. This includes rules that govern how sensor nodes should increase sampling rate in the event that a neighboring node fails. Nodes are also monitored to check when neighbouring nodes are transmitting correlated data. There are rules that put the other ones to sleep or reduce the sampling rate when this occurs. Once decisions are made, they are passed on to candidate nodes for implementation.

The candidate nodes also have a local catch where they store adopted rules for local adaptation so that it does not always consult or wait to be informed about previously encountered scenarios. Candidate nodes have context collectors at their application level responsible for collecting context information. Information about context is sent to the cluster header which then performs context processing, duty cycling and sending, broadcasting, and routing information.

Context collectors are software entities that provide raw data about the environment in which the sensor nodes are operating. This information comprises the node's resources, environmental parameters. This information in used in two ways: to enable context processing and routing decisions through the SCAEDM algorithm which we describe in section 4.4

4.4 SCEDM Algorithm.

The SCAEDM algorithm utilizes information gathered from many cross layer context information such as link quality, geographic progression, residual energy and rate of energy consumption to select forwarders which are responsible for relaying data packets to other potential forwarders until the data reaches its intended destination. The model also schedules sleep intervals of WSNs nodes according to the rate of energy consumption and the traffic load. The Stochastic context aware-energy consumption and distribution model(SCAEDM) integrates stochastic optimization with Geographic opportunistic routing and context-aware modeling ideas.

4.4.1 Packet Forwarding in SCAEDM

A node intending to transmit data initiates the transmission process by broadcasting the data packet. This packet comprises the data to be transmitted as well as the node's geographic location and the intended destination location. After the packet has been broadcast, several other neighboring nodes which we may term candidate forwarders overhear the broadcast transmission.

When the candidate forwarder overhears the transmission, they check their packet headers to determine if they can improve the geographic progression of the packet compared to the previous node. If a candidate forwarder does not improve the geographic location, it drops the packet. If a candidate forwarder improves the geographic progression, it starts a timer called Stochastic Energy Distribution Function (SEDF) and waits for the timer to elapse. The SEDF is derived from various cross layer contextual information and it allows the model to select the next hope forwarders, and hence distributes energy consumption among the various WSNs nodes. When the timer elapses, the candidate forwarder adds its geographic location information to the header information of the data packets and then rebroadcast the data packet. This process is repeated until the data packets reach the intended destination.

If a candidate forwarder overhears the forwarding of the same packet while its timer is running, it cancels its own timer as this implies a neighbour's timer has elapsed first and is hence the chosen forwarder.

Passive acknowledgement occurs when the original sender overhears the forwarding of data packets that it originally broadcast. This serves to inform that a neighbor has overheard the packet and is now retransmitting it to the next hop forwarders. Sometimes the original sender may overhear many passive acknowledgements from different neighbours. In such scenarios, the senders only accept the first response and ignore and ignore all the others.

When the first sender receives acknowledgement from a forwarder, it responds by establishing a unicast connection which is then used to transmit the rest of the data packets to the forwarder. The duration of this unicast transmission depends on quality of the link between the sender and the receiver.

$$SDFD = (w_1 \times LQM + w_2 \times \boldsymbol{Progression} + w_3 \times ECR + w_4 \times MinedC) \times SDFD_{max} \quad (43)$$

where w_1, w_2, w_3 *and* w_3 *and* w_4 are constants associated with link quality, progression, ECR and MinedC. MinedC is a variable representing additional mined contextual information such as sensor mobility. ECR is the energy consumption rate. $SDFD_{max}$ is a predefined maximum delay at each node. $w_1 + w_2 + \ w_3 + w_3 + w_4 = 0$. The metrics link quality, progression, ECR are discussed below. Transmitted Link Quality Metric (LQM): SCAEDM uses instantaneous link quality metric as measured after the transmission of data and this is measured as Received Signal Strength Indicator (RSSI).

$$LQM = \begin{cases} 1 & If\ RSSI_{transmitted} < RSSI_{threshold} \\ \frac{RSSI_{max} - RSSI_{transmitted}}{RSSI_{max}} & if RSSI_{threshold} < RSSI_{trasmitted} < RSSI_{disired} \\ 0 & if\ RSSI_{transmitted} > RSSI_{disired} \end{cases} \quad (44)$$

$RSSI_{transmitted}$ denotes measured real time received signal strength indicator, $RSSI_{threshold}$ denotes the minimum allowed signal strength indicator, and $RSSI_{max}$ denotes a predefined maximum value for received signal strength. Nodes with good link quality ($RSSI_{transmitted} > RSSI_{disired}$) contribute zero input to the LQM. This implies that they also contribute no input to the delay function SDFD and hence maximize their chances of being selected as the forwarding function since their timers have lower values and these are likely elapse earlier. On the contrary, nodes with bad links, $RSSI_{transmitted} < RSSI_{threshold}$ will contribute 1 towards the quality link metric indicating that they will contribute significantly to the delay function SDFD. This contribution results in a longer SDFD delay function and it jeopardizes the likelihood of the nodes being chosen as forwarding candidates. Neighbourhood selection and forwarding candidates were selected using algorithm 5.1, Candidate Selection and Neighbour Extraction.

Algorithm 5.1: Candidate Selection and Neigbour Extraction

$C \leftarrow EXTRACT\_NEIGHBOURS(N_S)$
  $F_s \leftarrow Include\_Candidates(C)$
  $SDFD_s \leftarrow s.computeSDFD(F_s)$
Return F_s
  $C \leftarrow N_s$
while $C \neq \emptyset$ do
  $SDFD_s \leftarrow s.computeSDFD(C)$
  $c_i \leftarrow EXCTRACT\_MAXIMUM(C)$
if $SDFD_S \leq OEC_{c_i}$ then
$C \leftarrow C - \{c_i\}$
Else
  return C
end if
end while

4.4.2 Geographic progression

We used the following formulae to compute the geographic progression for each potential forwarding candidate note:

$$progression = \frac{Ecludian\ distance(d_{x,y} - c_{x,y})}{Ecludian\ distance(d_{x,y} - s_{x,y})} \qquad (45)$$

Where $d_{x,y}$ denotes the geographic position of the destination node, $c_{x,y}$ denotes the geographic position of the candidate forwarding node, and $s_{x,y}$ denotes the geographic position of the source node. The Euclidian distance between any two points $P = (p_1, p_2)$, and $q = (q_1, q_2)$ is defined as:

$$Ecludian\ dist(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}. \tag{46}$$

Forwarding candidates that are too close to the source node but in the opposite direction to that of the destination node will have progression values that are greater than one. This will result in far much bigger values of SDFD and this makes them unlikely candidates for forwarding to the next hops. Nodes that are closer to the destination will have smaller progression values and this results in small contributions to the delay function SDFD, making such nodes more likely to be chosen as forwarding candidates, provided they are within threshold transmission range of the broadcasting node to enable overhearing.

4.4.3 Energy Consumption Distribution Function

The ECDF is defined as:

$$ECDF = \frac{RE_t}{ECR_t}, \tag{47}$$

where RE_t is the residual energy after t transmissions and ECR_t is instantaneous energy consumption rate. We used the weighted exponential smoothening function to estimate the residual energy (RE_t) remaining on each node after t transmissions. The residual energy $\widehat{e_t}$ each node after t transmissions is estimated by:

$$\widehat{e_t} = \widehat{e_{t-1}} + \propto (e_{t-1} - \widehat{e_{t-1}}), \tag{48}$$

where $\widehat{e_{t-1}}$ is the previous residual estimate, e_{t-1} is the actual previous residual energy and $\propto$ is a domain constant that determines the importance of the difference between the previous actual value and the previous estimate of residual energy.

4.5 SCEADM Context-aware-Duty Scheduling Algorithm

5.2 Context-aware Duty Scheduling Algorithm

```
Let EC = Energy consumption, ET = Threshold energy, and
TR=traffic rate.

IE EC > ET_2 then S_2 → S_5
end if
if ET_1 < EC < ET_2 then S_2 → S_4
end if
if EC ≤ ET_1 then
        if T_1 < TR ≤ T_2 then S_2 → S_1
end if
If T_2 < TR ≤ T_CSMA then S_2 → S_2
end if
if TR ≤ T_1 then S_2 → S_0
end if
end if
```

4.6 SCEADM Mining Algorithm

Consider a homogenous network with designated node D which is assigned the responsibility to receive reading from a group of sensor nodes in its neighborhood. Let $S = \{S_0, S_1, \dots, S_n\}$ be the set of nodes in this group. Let $R = \{r_0, r_1, \dots, r_m\}$ be the set of attributes of any sensor $S_i \in S$, $S_i \neq S_j$ for $0 \leq i, j \leq n, r_x = r_y$ for $0 \leq x, y \leq m$, the attributes r_x may represent residual energy, channel quality, spatial location, or other sensing attributes on which data may be collected. All attribute values of a sensor network node $S_i \in S$ may be denoted by $S_iA = S_0r_0, S_1\ r_1, \dots, s_ir_m$. Let $SA =$

$\{x_0, x_1, \dots x_t\}$ denote the set of all attributes for sensor nodes in the WSN. The data sensed by attributes r_x of sensor nodes S_i may be discretized into M levels to reduce processing requirements during rule construction.

$\forall s: S_i \in S$ collect context information about applications available on the application. Estimate approximate energy requirements for each application to run for a specified duration α.

Check_residual_energy_Update()
if is up_dateavailable
Else estimate residual energy.
While((residual energy ≥ α)

If memory cache for rules to fire not empty {

Context scheduler schedules application to run by selecting parameters for duty cycling based on fired rules.
Adjust sampling rate based on fired rules.
Adjust number of hops from which data is sent depending on context.
Route data using DVRD }
Else
{
Perform duty cycling based on parameters stored in context adapter
Adjust sampling based on sampling model and sampling parameters stored in context adapter.
Route data using
}
Adjust mobility
Transactions may be processed in batches $b_0, b_1, \dots, b_l$, where $b_i (0 \leq i \leq l)$ where i is large enough such that b_i cannot occur with equal probabilities. Let $T = t_0, t_1, \dots t_p$ be the set of transactions in b_i. Any transaction t_k, where $0 \leq k \leq p$, has the form

$\{S_0\, A, S_1, \ldots, S_n; count(t_k)\}$ where count is the frequency in time units in which any attribute in which each sensor attribute remains unchanged. Let $T_{test}(S_i r_x = \{tm_0, tm_1, \ldots tm_s\} \in T$, where $0 \leq s \leq p$ be the set of transactions in a batch b_i that contains the attribute $S_i r_x$ with value X. For any batch b_i, the support of any transaction $t_k \in T$ is given by:

$$support(t_k) = \frac{count(t_k)}{\sum_{j=0}^{r} count(t_i)} \tag{49}$$

$$\boldsymbol{support(s_i\, a_x = X) = \frac{\sum_{i=0}^{r} count(tmv_i)}{\sum_{j=0}^{s} count(tm_j)}} \tag{50}$$

4.2.3 SCEADM Data Mining Algorithm

1: BEGIN

2: Set *frequentTransactionscount* List to store to store counts for frequent transactions in the list.

3: $Set: frequent\ Transaction\ count\ (t_i)$: this represents the count for frequent transactions and mostFrequentTransaction to denote the most frequent Transaction in *FrequentTransactions.*

4: set maxSupport as current maximum support.

5: end declaration

6: obtain energy levels of sensors:

$Energy\ S = e_0, e_1, .., e_q.$

7: Generate covariance Matrix, C_m.

8: Using a bitmap, initialize two sensor components within S with the highest probability measure from, C_m and Energy S.

9: Convert the continuous values in b_j into discrete values.

10: for *i*=1 to |*frequentTransationsCount(t_i)*| do

11: $currentSupport = \frac{frequentTransactioncount(t_i)}{|b_j|}$

12: $\boldsymbol{if}\ currentSupport > maxSupport\ \boldsymbol{then}$

13: $maxSupport = currentSupport$

14: end if

15: $\boldsymbol{if}\ highestSupport \geq thresholdSupport\ \boldsymbol{then}$

$R = getRules\ (mostFrequentTransaction)$

else if number of bits > 2 then

if all bits set then

Reset all bits to zero

else

Remove one bit reflecting current highest correlation matrix

end if

4.3 Comparison with similar work

The novelty of our approach lies in the choices made in the architecture of our SCAEDM model framework and algorithms. While our approach is similar to other works in [56] [57] that it uses context information to regulate energy consumption, there are major differences in the manner in which the context information is obtained and utilized. Previous works have focused on using context information to minimize energy consumption through network metrics such as average energy consumption or total energy consumption of the network. Some of these approaches focus on the use of task scheduling using a fixed set of pre-existing rules within some knowledge base. Our approach focuses on reducing energy consumption at the node level and hence network level. While our approach also utilizes pre-existing rules in a knowledge base like some of the previous works, the focus is on integrating these with new mined rules based on context applications services data. Furthermore, our methods optimize on routes selecting through the use of context factor metric.

4.4 Chapter Summary

The chapter presented a stochastic context-aware energy consumption and distribution model for WSNs. Initially, we presented the framework for the model and the various processes underlying the model such as context processing, context adaptation, routing and media access. We discussed how the various model components interact.

The chapter also presented and discussed the SCEADM algorithm. We discussed how candidate nodes (nodes that forward data packets) are selected and how packet forwarding is accomplished in SCEADM. We discussed how the SEDF, a function that is used to select forwarding candidates is computed using various context parameters such as residual energy, channel quality, geographic progression etc. We also discussed how the computation of metrics for computing SEDF, such as residual energy, channel quality, geographic progression etc. are performed. We presented and discussed the algorithm for selecting forwarding candidate nodes.

The chapter also presented and discussed Context-aware Duty Scheduling algorithm which is responsible for node transitions into various transition states that enable the network to perform its functions as well as putting nodes to sleep in an energy efficient manner. Finally, we presented and discussed the SCAEDM algorithm responsible for mining context rules and adapting the nodes to the contextual environments.

Chapter 5

Results, Evaluation and Implications

5.1 Chapter Overview

Chapter five discusses how the concept of Field programmable gate arrays could be exploited to design reconfigurable smart sensor systems. This chapter therefore provides a comprehensive evaluation methodology of the work covered in this thesis and provides a platform to compare the works with current research trends on smart sensor design.

FPGA provides power efficient circuits that are architecturally flexible due to their parallelism. Of late, FPGA have improved the performance of smart devices and this has triggered interest in their application to the design of smart sensor cores. The current market leaders in the FPGA industry are Xilinx and Altera. They have developed FPGA architectures that allow implementation of combinational and sequential circuits, with capabilities for inclusion of soft high-level soft processors [99]. The rationale behind these soft processor cores is to enable designers to customize designs, reduce costs as well as enable hardware acceleration.

However, devices implemented using these FPGAs exhibit different characteristics due to differences in manufacturers as well as the specifications used in FPGA. Thus, current trends in the development of smart sensor systems require appropriate benchmarking methodologies in order to characterise the desirable features and attributes of smart sensor devices such as chip size, accuracy, resolution and power consumption requirements for the various reconfigurable architectures to be comparable.

5.2 FPGA Device Evaluation approaches

When selecting an evaluation method for FPGA devices, it is important to select those that are application independent. Moreover, a suitable evaluation method should allow for fast, easy evaluations capable of enabling designers to evaluate designs for architectural improvements. The evaluation method should also be applicable to a wide range of devices. The power consumption evaluation of devices is, to a certain extent, influenced by factors such as implementation tools and the measurement environment. It is also important that the evaluation methods enable fair comparison and that the results are independent of the implementation tools used [100]. It is therefore beneficial to use benchmarking techniques to minimize the influence of the implementation tools to some extent. Another characteristic of a good evaluation method is that it should allow the designer to capture different power levels \ modes and enable the possible addition of future techniques currently unavailable.

The evaluation methods used in this research are Power Estimation methods using FPGA, in our case the Power Analyser tool Altera model Sim and a power benchmarking suite for reconfigurable architectures, GroundHog 2009 [101].

5.2.1 Power Estimation using FPGA CAD

One of the major approaches in optimizing an FPGA device for low power consumption is to map the particular application onto a range of target devices and then evaluate the device power consumption using power estimation tools provided by the FPGA. Some of the major drawbacks of this approach include limited accuracy and lack of standards across different devices and manufacturers.

5.2.2 FPGA Characterisation Benchmark

The approach implements a highly active circuit on FPGA in order to measure power consumption in various activity modes. The activity modes considered are active processing, inactivity and dedicated low-power states. The method used comprises the following:

i. A facillity for random number generation to test high activity circuitry.

ii. High level usage of logic resources in device.
iii. Running the evaluation circuit at fixed clock rate (100 MHz) when active.
iv. Identifying activity modes and switching between them.
v. Measurement of power and temperature in the various modes.

The power measurement setup consists of the following components: the power measurement systems, the stimulus generator and the design under test (DUT) as indicated in Figure 5.1. The power measurement system is used to measure power consumption of the design under test unit. This may be achieved by either measuring the core voltage or the current drawn by the core. When only current is considered for measurement, it is necessary to stabilize the core voltage to a fixed level in order to ensure that the voltage being supplied does not drop when the DUT draws large amounts of current. The purpose of the stimulus generator is to generate workload (stimulus signals) which is then passed to the DUT. In our evaluation, the stimulus generator was implemented on FPGA which was then connected to DUT.

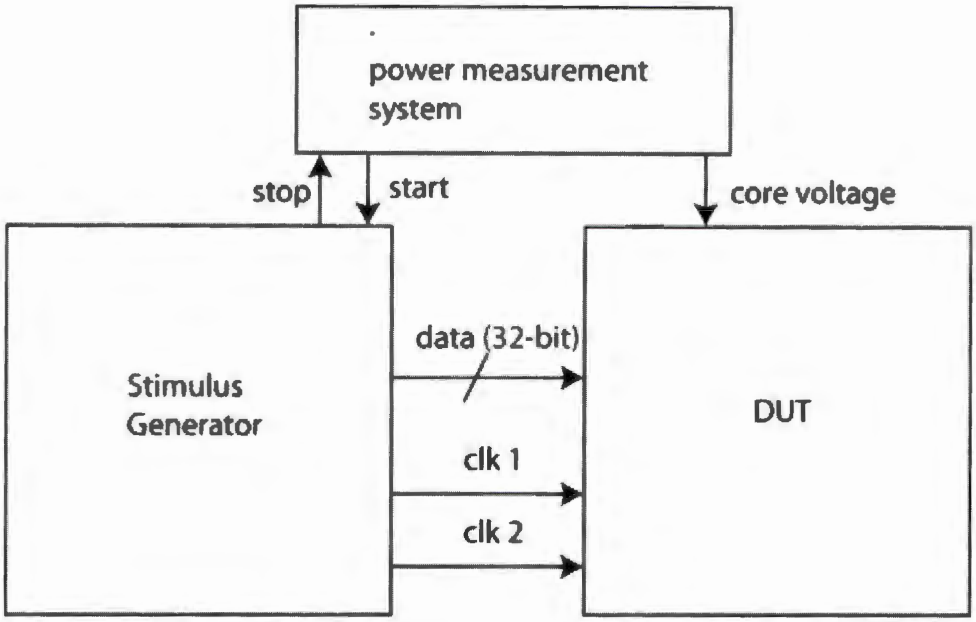


Figure 5.1 Power Measurement System

5.3 MOGAFPGA Design Evaluation

Table 6.1 shows details of groundhog device characterization experiments for various Xilinx devices upon which this research was benched marked. The table lists various lower power devices and their corresponding energy consumption characteristics such as core voltage, thermal resistance, number of Luts and logic utilization. The research compares the energy consumption characteristic of Xilinnx Vertex-II XC2VP30 and Xilinx Vertix-5 XC5VLX50T device with our Multi-Objective Genetic Algorithm Field Programmable Gate array design (MOGAFPGA).

Table 5.1: Groundhog Fabric Characterization

FPGA family	Device	Board	Process Technology	Core Voltage	Thermal Resistance	Number of Luts	Number of Rings	Logic Utilisation
Xilinx Vertex -II	XC2VP30	XUP	130 nm	1.5 V	0.6°C/W	26.4	46	89.9
Hilinx Spartan-3E	XC3500E	SPARTAN-3 Starter Kit	90 nm	1.2 V	9.8 °C/W	9.312	16	88%
Xilinx Spartan-3AN	XC3S700AN	Spartan-3AN Starter kit	90 nm	1.2 V	5.3°C/W	11.776	21	91.3
Xilinx Vertix-5	XC5VLX50T	ML505	65 nm	1.0 V	0.2 °C/W	28.800	50	88.9
Silicon	ICE65I04	ICEman Eval Kit	65 nm	1.2 V	n/a	3.520	6	83.7

Table 5.1 shows MOGAFPGA design optimisation output for various evaluation parameters such as total logic elements, total combinational functions, dedicated registers total virtual pins and embedded multiplier cores. The table shows generally lower percentage utilization of the components compared to those of the groundhog benchmark experiments. This implies that MOGAFPGA design generally requires fewer components to perform the same tasks than would be required by other devices in the groundhog characterization experiments.

5.4 Simulation Results for MOGAFPGA

The power dissipation of FFT algorithm was determined using PowerPlay Analyser tool available in Quartus II synthesis tool. These tools perform power analysis on data obtained after synthesis using VCD file containing signal activities of FFT core. The functionality of individual components was verified by running test benches in ModelSim (simulation tool for Mentographics Inc.). The FFT processor algorithm as a complete system was functionally verified using system level test bench tools in

Modelsim. The model was also evaluated in terms of functional correctness and portability. The results show significant improvements in all performance measures.

Programmable logic units, (Luts-representing genes in a chromosome) were designed using schematic /block diagrams and hardware description language and they were encoded into chromosomes and then optimized using field programmable gate arrays (FPGA) to achieve desired goals of portability and low energy consumption. The performance of our design was evaluated using Altera Quartus II Powerplay power analyser tool in terms of core static thermal power dissipation, core dynamic thermal power dissipation and input and output thermal power dissipation. Experimental results reveal that our genetic algorithms improved the design of the desired reconfigurable circuits in terms of energy consumption and device portability in a few generations.

Figure 5.2 shows Powerplay Power analysis summary results for sample FPGA. The Powerplay Power Analysis tool enables us to estimate the power of a device accurately. Electrical energy supplied by external power sources is needed for proper external and internal operation of FPGA devices. Designers need to understand the total power required from these supplies when implementing power supply solutions. Moreover, designers need to consider how much of that total power is actually dissipated within the device (referred to as "thermal power" or "dissipated power") compared to the proportion of power that is dissipated outside the device, such as external output capacitive loads and balanced resistor termination networks.

Thermal power is the power that dissipates as heat from the FPGA. For a good device, this energy must be sufficiently low to allow a cooling solution for the device. The total power consumed by a device is comprised of standby power, dynamic power and I/O power components. Standby power is from current in the device in standby mode. Core dynamic power is from internal switching within the device (changing and discharging capacitance on internal nodes) [1]. I/O power is from external switching (charging and discharging external load capacitance connected to device pins), I/O drivers and external termination network.

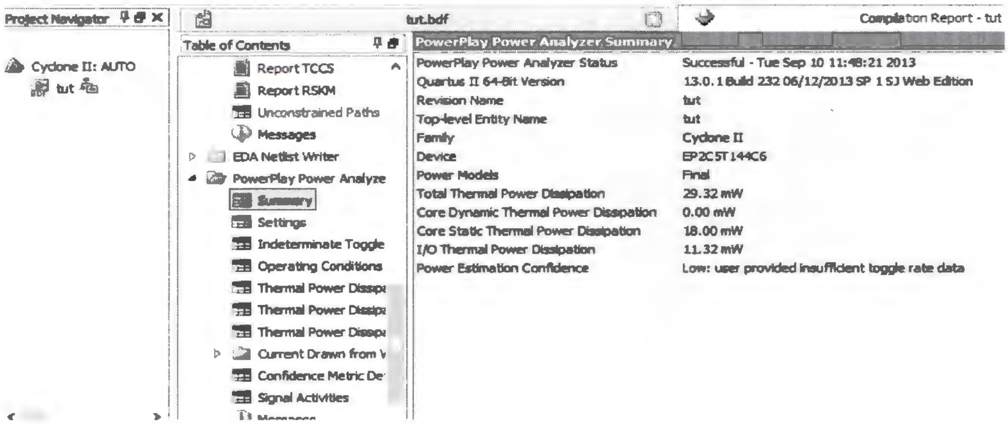


Figure 5.2 Sample Powerplay analysis summary for simple programmable logic units

The results of Figure 5.2 show Total Thermal Power Dissipation of 29.32 mW, Core Static Thermal Power Dissipation of 18 mW, Input/output Thermal Power Dissipation of 11.32 mW and zero Core Dynamic Thermal Power Dissipation.

An analysis of Total Thermal Power consumption with population evolution was carried out for our integrated reconfigurable circuit device. The results in Figure 5.3 show that the Total Thermal Power consumption generally decreased as the circuit evolved. This demonstrates the ability of our genetic algorithm to improve the performance of the circuit design under consideration.

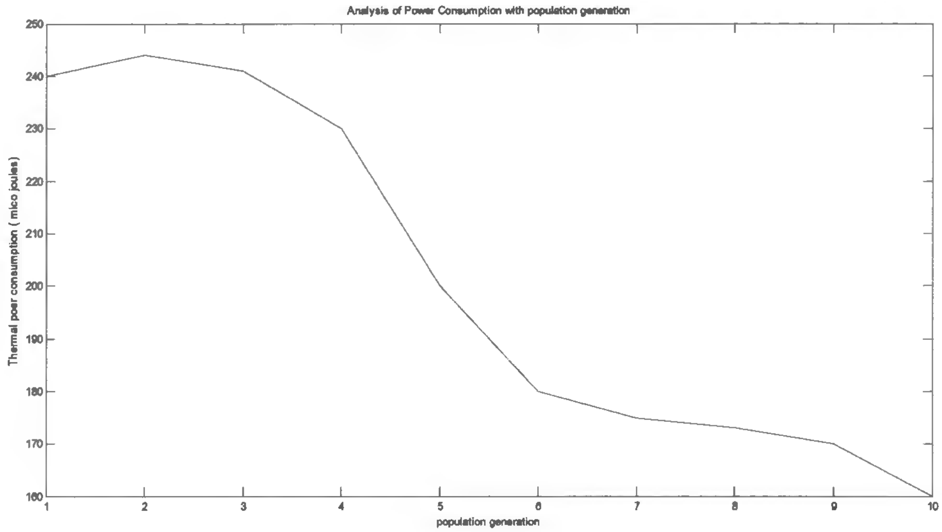


Figure 5.3: Thermal power consumption against population generation

An analysis of Total Thermal Power consumption with population evolution was carried out for our integrated reconfigurable circuit device. The results in Figure 5.3 show that the Total Thermal Power consumption generally decreased as the circuit evolved. This demonstrates the ability of our genetic algorithm to improve the performance of the circuit design under consideration [1].

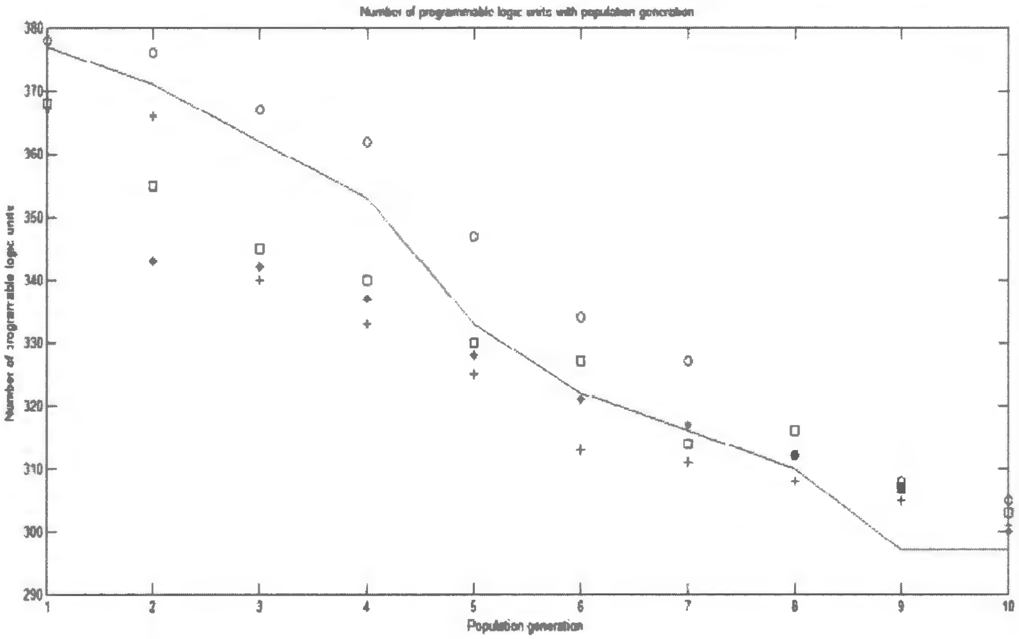


Figure 5.4 Number of programmable logic units with population generation

Figure 5.4 shows an analysis of the number of programmable logic units with population evolution for the integrated circuit design. The results show that our algorithms managed to significantly reduce the number of logic units required to perform the task at hand. The reduced number of logic units in turn leads to reduced energy consumption as reported above. Furthermore, this reduction in the number of logic units in turn leads to reduced number of FPGA slices as indicated in Figure 5.5. This reduction in FPGA is interpreted as a good measure of the portability of the physical device when implemented in hardware.

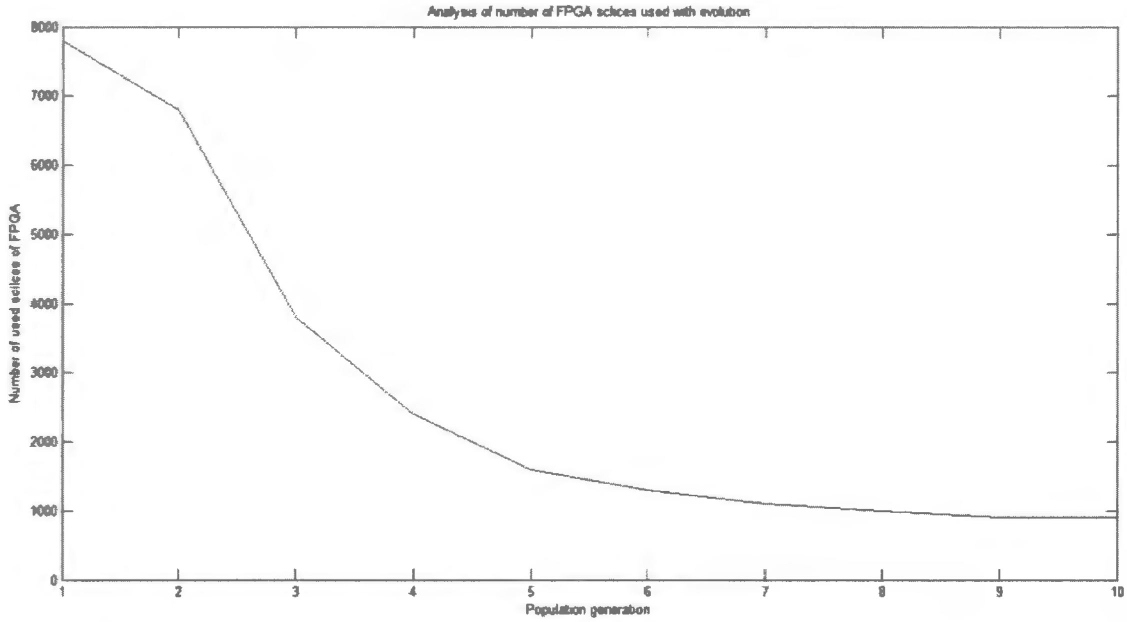


Figure 5.5 Number of FPGA slices with Evolution.

Table 5.2: MOGAFPGA Device Optimisation

Target information

Vendor: Altera

Family: Clone II

Device: EP5CST144C6

Optimisation Goal: Component Usage

Evaluation Parameter	Optimization Output	Percentage
Total Logic Elements	1335/4608	28.9 %
Total Combinational Functions	1310/4608	28.4 %
Dedicated Registers	8/30	26.6 %
Total Pins	16/89	17.9 %
Total Virtual Pins	8/20	40 %
Total Memory Bits	18250/119808	15.2
Embedded Multiplier 9bit Elements	8/26	30.8
Total PLLs	¼	25

Table 5.2 shows a comparison of device surface temperature for MOGAFPGA design with V5XL50T and V2P30 low power device cores. Compared to MOGAFPGA, V2P30 device reaches surface temperatures of over 70°C and it is capable of overheating. In contrast, MOGAFPGA and V5XL50T are immune to this problem as they are not capable of reaching over 50 °C.

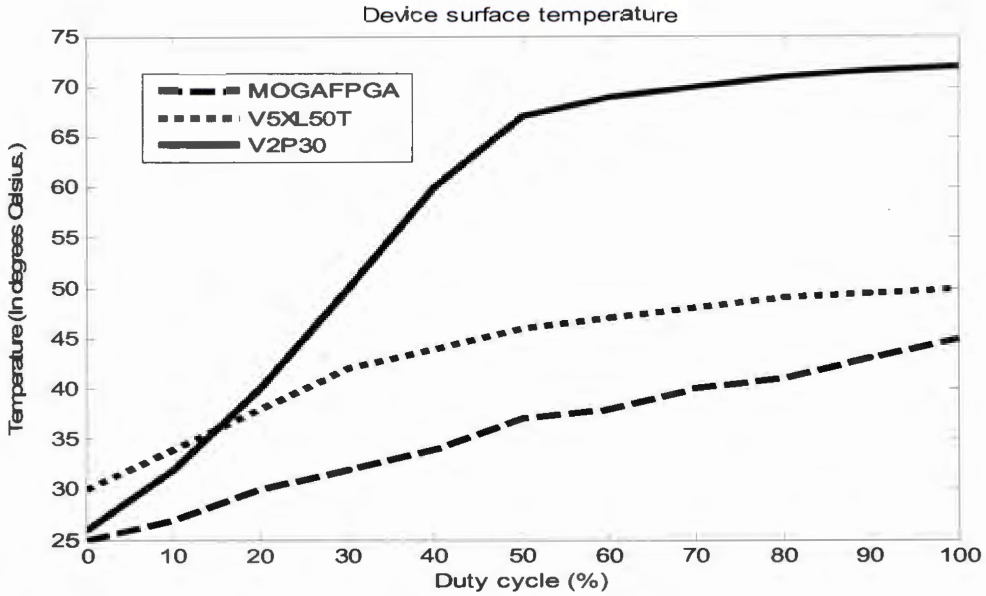
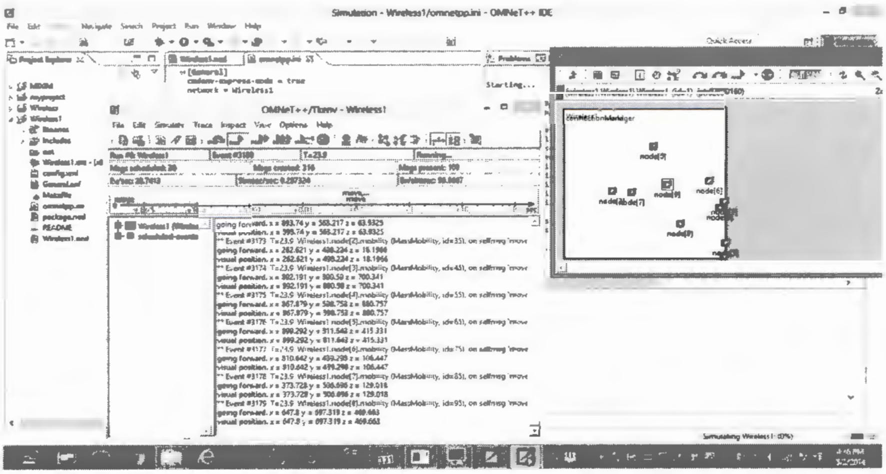


Figure 5.6: Device surface temperature

Figure 5.6 shows how inactive power (similar to static power) consumption varies with duty cycle. Figure 5.6 illustrates a general increase of inactive power with increase in duty cycle. This behaviour may be attributed to increases in temperature of the device generated during active parts of duty cycle.

5.5 Simulation and Analysis of Results for SCEADM

Comprehensive simulations were carried out in OMNet++ framework using Mixim (Mixed Simulator) simulator. The results show a normally distributed general increase in number of packets transmitted with increase in number of sensor nodes. Simulating wireless communications systems require suitable abstractions of the environment, the radio channels and the physical layer. Simulation was carried out on an OMNeT++ framework using MiXim (Mixed Simulator) simulator. OMNeT++ provides a base framework for general functionality needed for wireless simulations such as connection management, mobility and wireless channel modelling.



5.7 Communication Diagram

Mixim simulator is an extension of various wireless and mobile simulation Frameworks in OMNeT++. It is mostly used for simulating wireless channel models and communication and mobility protocols such as Medium Access Control (MAC) level. Figure 5.8 shows a sequence chart depicting energy consumption and the communication process for four sensor nodes. The transmissions along the horizontal axis represent the sending of messages by the nodes while the height of the nodes represents residual energy of nodes.

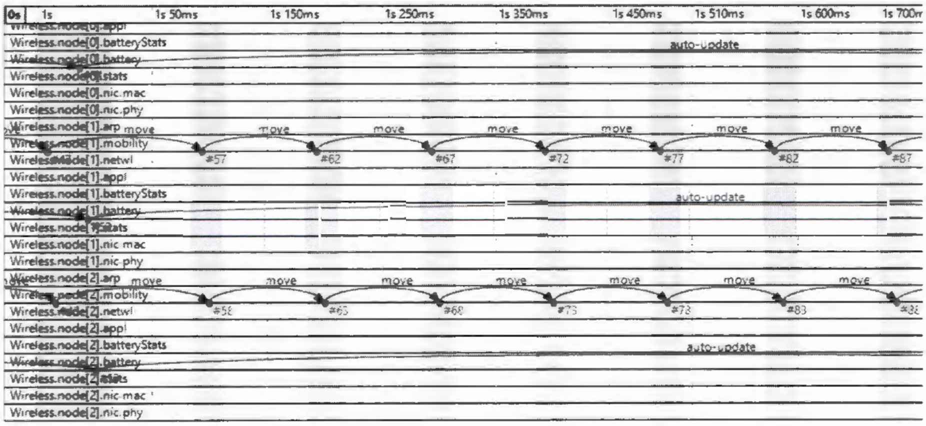


Figure 5.8 Communication Sequence Chart

We carried out several simulations runs to investigate the relationship among several simulation parameters in our model. Figure 5.9 shows such a simulation run displaying the global traffic, usage, number of frames sent, number of frames received, number of dropped frames, number of mixed acknowledgements and number of dropped acknowledgements etc.

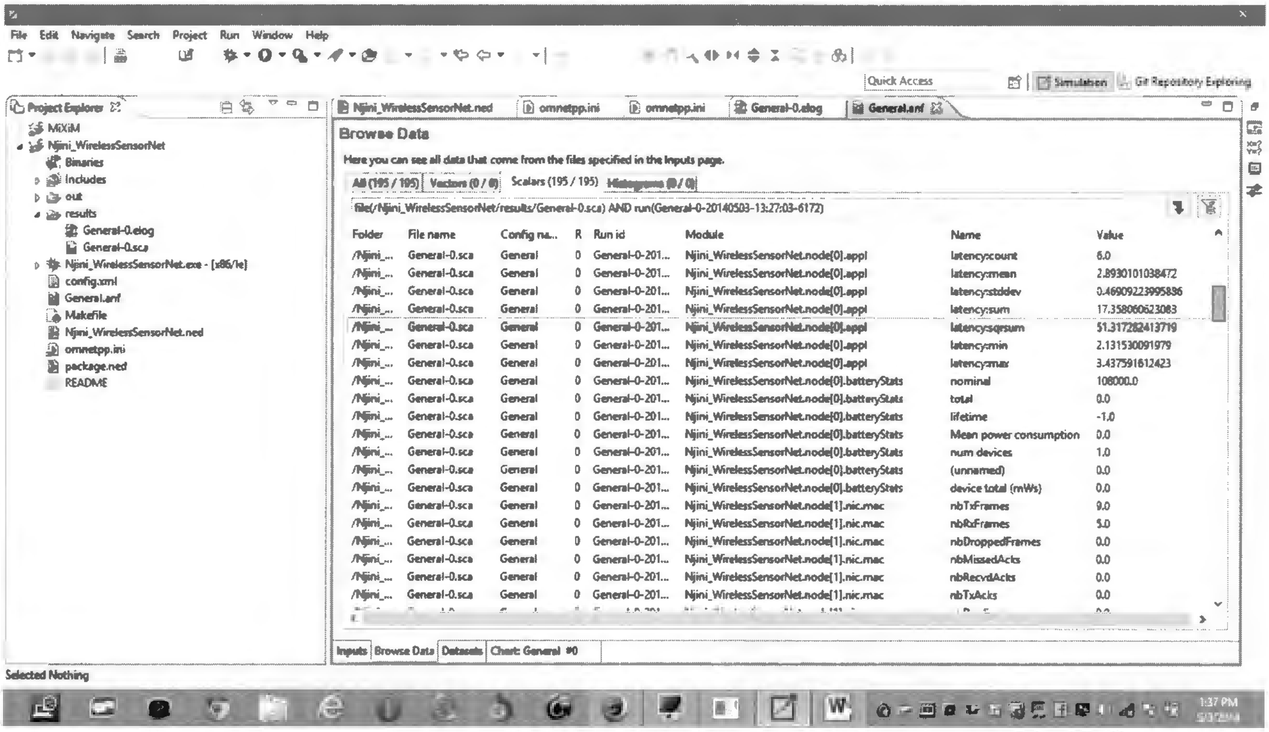
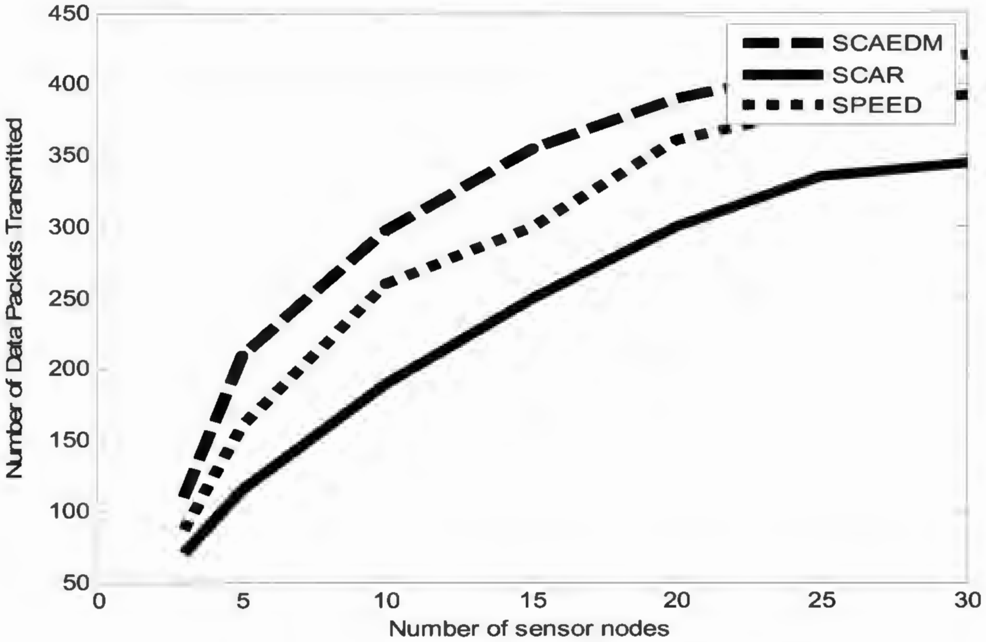


Figure 5.9 Sample simulation run depicting various model parameters

An empirical analysis was carried out to investigate how transmission may be affected by the number of sensor nodes present in the network. Simulation runs were carried out for nodes 3, 5, 10 up to 30 as indicated on Figure 5.10. The results show a normally distributed, general increase in number of packets transmitted with increase in number of nodes in the network. The results show a marked improvement in the number of packets transmitted compared to similar works such as SCAR and SPEED.



5.10: Transmitted Packets

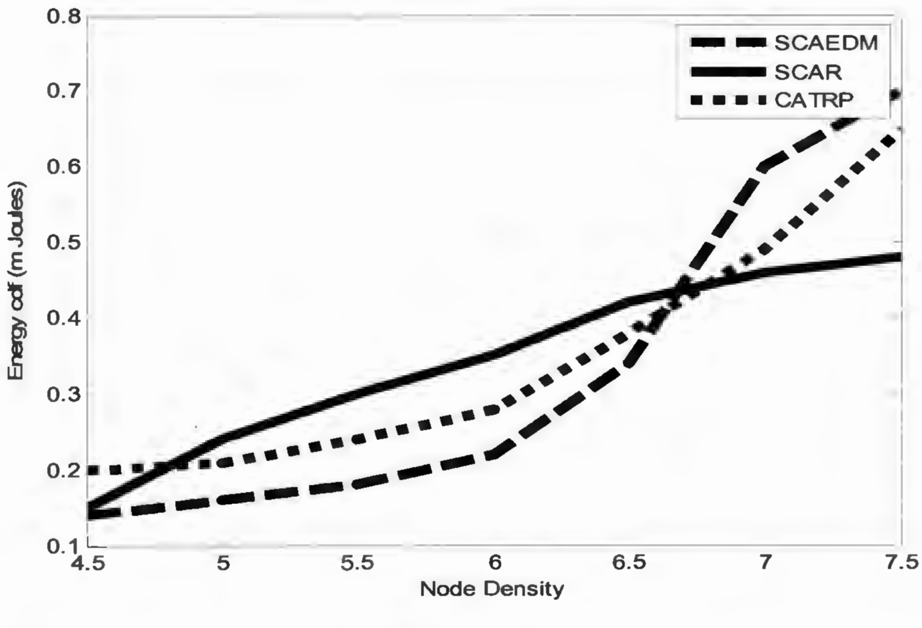


Figure 5.11 Cumulative distribution function of Energy

Figure 5.11 shows the results of empirical analysis of the *cdf* of the energy consumption during the 10 minutes for various node densities. The results show a general increases in cdf with increase in node density. This means that in areas where the nodes were highly populated, the nodes tended to retain energy than the lower density of nodes.

5.6 Impact of Node density

We compare SCAEDM with EQGOR and Multipath Routing Protocol (MPQR). The performance of SCEADM for different node densities is evaluated. We evaluate this performance by recording the number of received packets for various node densities ranging from 100 nodes to 200 nodes. The Simulation results are shown in Figure 5.12. The results show that SCAEDM and EQGOR achieved shorter end-to-end delivery delay than MPQR. This is because both SCEADM and EQGOR both use packet progression relay priority rules.

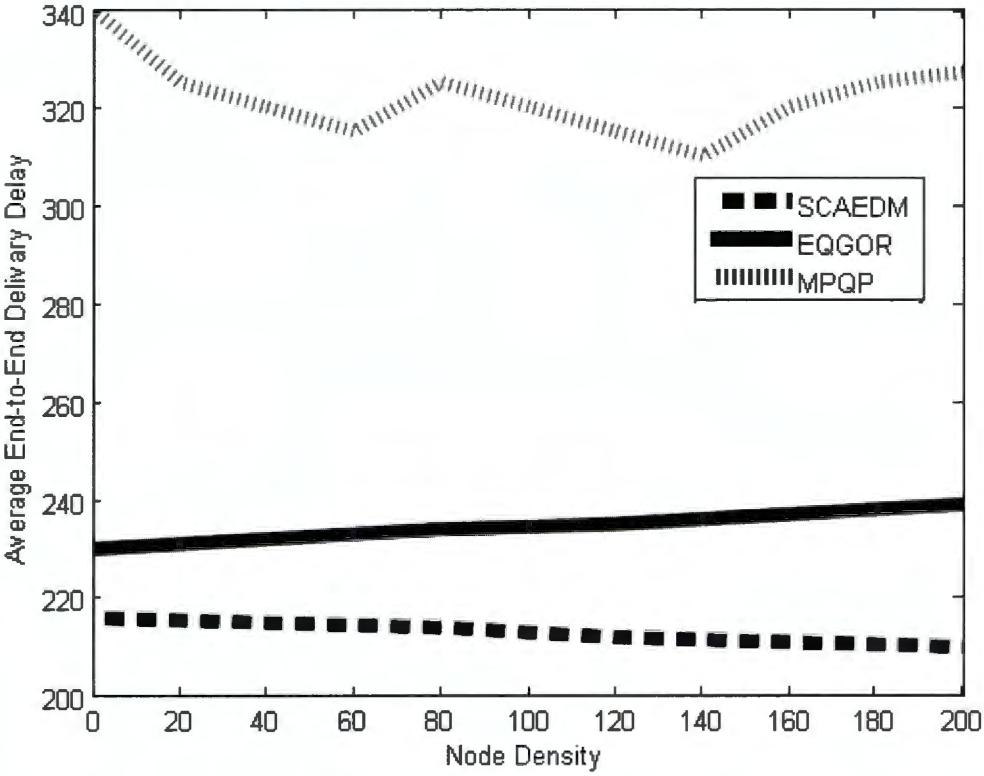


Figure 5.12: Impact of Node Density

The results also show that SCAEDM out performed EQGOR. This better performance of SCAEDM is attributed to increased sensitivity to context information by the algorithm which is achieved through more accurate data collection and improved parameter estimators such as those used for estimating residual energy and channel quality. Moreover, the results show that the end-to-end delay for MPQP is adversely affected by the impact of node density. This is mainly caused by the fact that multi-path routing increases channel contention which significantly reduces end-to-end delay performance. We also observe that end-to-end delay for SCEADM decreases further as the node density increases. This is because increasing the node density leads to an increase in potentially good forwarding candidates and hence better progression.

5.7 Chapter Summary

The chapter reviewed the application of stochastic models in energy consumption analysis. The concept of context-aware computing in WSNs is integrated with the theory of stochastic process to build a stochastic context- aware energy consumption and distribution model for multi-hop WSNs (SCAEDM). Our model is capable of mining, discovering and capturing multiple context parameters in multiple dimensions of the network stack. The results show exponential *cdf* for various selected node densities from 5, 10, 15, to 30 nodes. Sensor nodes in our model only collect and transmit data periodically, thus allowing for energy conservation through reduced node activity.

Chapter 6

Summary, Conclusion and Future Work

6.1 Summary

The research study outlined problems associated with energy efficiency for WSNs. The research discussed the nature of problems that occur and the context in which these problems occur. The research further provided the research questions and objectives that helped to shape and focus the research process.

The main problem that was explored is how to minimize the energy utilized by the sensors in order to extend the life of the sensor network. This problem is further exacerbated by the limited communication range of radio communication used by the sensor nodes. Our literature survey also indicated that the bulk of WSNs energy consumption is due to ineffective node communication resulting in poor load balancing. Furthermore, the bulk of energy expended during that ineffective node communication is associated with complex number multiplication. Yet, existing literature has failed to handle these issues of ineffective node communication resulting in poor load balancing and complex number multiplication which leads to increased energy depletion.

In Chapter two, the research reviewed existing alternative techniques and methods that may provide solutions to the problems raised. This research revealed how genetic algorithms have been used to model sensor communication, in clustering and routing problems. We also provided a performance evaluation of various GA-based optimization strategies. Our observations showed that while several algorithms try to select the best cluster headers or routing path based on some metric, the process normally introduces overheads in communication, which in turn leads to more energy dissipation.

Chapter two also reviewed the problems encountered in optimisation of multiple conflicting objectives as applied to WSNs. The chapter reviewed how genetic algorithms have been exploited to establish energy efficient WSN protocols. This was accomplished by initially discussing the multi-objective optimisation problem in its general form and then later discussing how it has been applied to optimisation of WSNs algorithms. The chapter also reviewed various load balancing approaches used for WSNs, focusing on Tree based load balancing and models based on Channel state information.

We reviewed several techniques that could be applied in WSNs in order to minimize the transmission of redundant data in order to minimize energy consumption. We discussed mainly the techniques that could be exploited to reduce the transmission of spatially and temporally correlated data in WSNs.

The major efficient techniques discussed are distributed source coding, which is an extension of the Slepian- Wolf coding technique, and Compressive sensing. We evaluated the use of signal reconstruction methods such as Convex Relaxation, Greedy Iterative Algorithms, Iterative Thresholding Algorithms and Bregman Iterative Algorithms.

In Chapter three we presented our novel technique through a simulated approach where we developed a solution by the application of Multi-Objective Genetic Algorithm for Field Programmable Gate Arrays in order to resolve the challenge of complex number multiplication during signal processing at the node level. In handling the challenge of ineffective node communication, resulting in poor load balancing between nodes, we designed a stochastic context-aware energy consumption and distribution model that effectively resolves the problem of load balancing.

In Chapter three, the research developed two models that helped to establish solutions to the research problem. The research contributed two models: MOGAFPGA and SCAEDM as part of the research contribution. Our initial approach was to reduce energy consumption at the node level. This energy is mainly consumed during signal transmission. Consequently, our novel technique is realized through a simulated approach where we developed a solution by the application of Multi-Objective Genetic Algorithm for Field Programmable Gate Arrays in order resolve the challenge of complex number multiplication during signal processing at the node level. MOGAFPGA is a multi-objective genetic algorithm that is implemented using Field Programmable arrays to address the problems associated with signal processing.

In handling the challenge of ineffective node communication, resulting in poor load balancing between nodes, we designed a stochastic context-aware energy consumption and distribution model that effectively resolves the problem of load balancing. We showed how candidate nodes (nodes that forward data packets) are selected and how packet forwarding is accomplished in SCEADM. Our novel contribution is the SEDF, a function that is used to select forwarding candidates. This function is computed using various context parameters such as residual energy, channel quality and geographic progression. We also discussed how the computation of metrics for computing SEDF, such as residual energy, channel quality and geographic progression were computed. We presented and discussed the algorithm for selecting forwarding candidate nodes. We also presented an algorithm for performing Context-aware Duty Scheduling algorithm which is responsible for energy efficient node transitions. Finally, we presented and discussed the SCAEDM algorithm responsible for mining context rules and adapting the nodes to the contextual environments.

6.2 Conclusion

The research focus was to develop energy efficient algorithms for computing Fast Fourier Transforms that reduce redundancy in processing operations, data duplication during data transmissions while achieving load balancing and overall energy conservation for the sensor network. The research performed experiments and simulations to demonstrate that it is indeed possible to design and implement algorithms and methods to achieve the objectives of the research. The research then provided an analysis and discussion of the results. SCEADM is a context aware energy consumption and distribution model that is intended to dynamically balance network load by considering runtime application demands and adjusting node behavior.

The research also provides direction of future work and current research trends in WSNs. The research then concludes by proving directions for future research in WSNs.

6.3 Future research

The limitation of this research is that the computation of FFT on FPGA using GAs is generally slow as it involves many computations and evaluations. Furthermore, in some instances, genetic algorithms have a limitation of being trapped in local optimum solutions. This problem could be resolved by implementing parallel Gas. We did not manage to use parallel GAs due to time constraints. In future research, we intend to explore the use of parallel Gas in computing FFTs using FPGA to further improve computational time and overall performances.

References

[1] I. Njini and O. O. Ekabua, "Multi-Objective Genetic Algorithms for computing Fast Fourier Transform for evolving Smart Sensors devices using Field Programmable gate arrays," *Advances in Computer Science: an International Journal,* vol. 3, no. 2, pp. 1-9, March 2014.

[2] I. F. Akyildiz, W. Su*, Y. Sankarasubramanian and E. Cayrci, "Wireless sensor networks: a survey," *Computer networks,* vol. 38, no. 4, pp. 393-422, December 2001.

[3] G. J. Proakis and G. D. Manolakis, Digital Signal Processing, Principes, Algorithms, and applications, 2, Ed., London: McGraw-Hill, 1968, pp. 253-265.

[4] P. Bergamo, S. Asgari, H. Wang and D. Y. Maniezzo, "Collaborative sensor networking towards real-time acoustical beamforming in free-space and limited reverberance," *IEEE Transactions on mobile computing,* vol. 3, no. 3, pp. 314-367, March 2004.

[5] J. P. Hong and N. Sulaiman, "Genetic Algorithm Optimization for Coefficient of FFT processor," *Australian journal of Basic and Applied Sciences,* vol. 4, no. 9, pp. 184-192, December 2010..

[6] M. Ayman, E. Khashab and A. M. Swartlander, " The modular pipeline Fast Fourier Transform algorithm and architecture," *IEEE,* vol. 499, no. 1, pp. 223-228, December 1998.

[7] R. R. Chiasserserini and R. Rao, "On the concept of distributed Digital signal processing in wireless sensors networks," *IEEE,* vol. 38, no. 6, pp. 266-266., March 2002.

[8] G. liang, X. Rui , T. Bunyuan and X. J. Jianping , " Data fusion improves the improves the coverage of wireless sensor networks," *ACM,* vol. 41, no. 7, pp. 212-230, Septermber 2009.

[9] W. Wng, C. Mehmet and V. Goddard, "Stochastic Analysis of Energy consumption in wireless sensor networks," *IEEE,* vol. 48, no. 3, pp. 255-278, september 2004.

[10] N. Nasser, C. Yunfeng Chen and N. Nidal, "SEEM: Secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer Communications,* vol. 30, no. 11, pp. 2401-2412, November 2007.

[11] I. F. Akyildiz, F. Ian, S. Weilian and S. Yogesh, "Wireless sensor networks: a survey," *Akyildiz, Ian F., Weilian Su, Yogesh Sankarasubramaniam, and Erda Computer networks,* vol. 38, no. 4, pp. 393-422, December 2002.

[12] O. Olariu, S. Stephan and S. Ivan , "Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting," *INFOCOM,* vol. 40, no. 6, pp. 1-12, December 2006.

[13] Stankovic, A. John, F. Tarek and C. Abdelzaher, "Real-time communication and coordination in embedded sensor networks," in *Real-time communication a Proceedings of the IEEE*, 2003.

[14] Knowles and D. Joshua, "Local-search and hybrid evolutionary algorithms for Pareto optimization," *Department of Computer Science, University of Reading,* vol. 25, no. 5, pp. 220-227, December 2002.

[15] T. Bäck, B. David and M. Zbigniew, "Evolutionary computation," *Bäck, Thomas, David B. Fogel, and Zbigni Basic algorithms and operators. Vol. 1. CRC Press, 2000,* vol. 10, no. 1, pp. 121-133, January 2000.

[16] Michalewicz and Zbigniew, " Genetic algorithms+ data structures= evolution programs," *springer, 1996,* 1996.

[17] Haupt, L. Randy and E. H. Sue, Practical genetic algorithms, 2 ed., John Wiley & Sons, 2004.

[18] Larrañaga, Pedro and A. L. Jose, " Estimation of distribution algorithms: A new tool for evolutionary computation," *Springer,* vol. 55, no. 5, pp. 223-229, July 2002.

[19] Eiben, E. Agoston , E. James and S. Smith, " Introduction to evolutionary computing," *Springer,* vol. 24, no. 7, pp. 332-343, July 2003..

[20] F. Falkenauer and E. Emanue, "A new representation and operators for genetic algorithms applied to grouping problems," *Evolutionary computation,* vol. 2, no. 5, pp. 123-144, January 1994.

[21] A. Nikolaos, A. Stefanos and N. Nikolidakis, "Energy-efficient routing protocols in wireless sensor networks: A survey.," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 2, pp. 551-591, 2013.

[22] S. Jin, Z. Ming and S. S. Annie, "Sensor network optimization using a genetic algorithm," in *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, 2003..

[23] J. Jia, W. Xueli, C. Jian and W. Xingwei, "Exploiting sensor redistribution for eliminating the energy hole problem," in *Jia, Jie, Xueli Wu, Jian Chen, and XinEURASIP Journal on Wireless Communications and Networking*, 2012.

[24] J. Loo, H. Jesús and L. M. Jaime, "Loo, Jonathan, Jaime Lloret Mauri, and J Mobile Ad Hoc Networks: Current Status and Future Trends," *Mobile Ad Hoc Networks,* vol. 6, no. 2, pp. 212-220, July 2012.

[25] J. M. Kim, H. P. Seon and J. H. Young, "KimCHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks," in *10th international conference on Advanced communication technology*, 2008.

[26] Bayraklı, s. Selim and Z. Senol , "Genetic Algorithm Based Energy Efficient Clusters (GABEEC) in Wireless Sensor Networks," in *Procedia Computer Science*, 2012.

[27] C. Fonseca, M. Carlos and J. F. Peter, "Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization," *ICGA,* vol. 93, no. 5, pp. 416-423, November 1993.

[28] C. Fonseca, M. Carlos and J. F. Peter, "Genetic Algorithms for Multiobjective Optimization: Formulation and Discussion and Generalization," *InICGA,* vol. 93, no. 8, pp. 416-423, July 1993.

[29] B. Deb and A. Kalyanmoy, "Introduction to evolutionary multiobjective optimization," in *Deb, Kalyanmoy. "Introduction to evolutionary muSpringer International Conference in Multiobjective Optimization*, Berlin Heidelberg, 2008..

[30] V. Martins, V. C. Flávio, G. Eduardo and E. Carrano, "MaA dynamic multiobjective hybrid approach for designing wireless sensor networks," in *IEEE Congress on Evolutionary computation*, CEC, 2009.

[31] K. EkbataniFard, G. Hossein, M. Reza and M. R. Akbar, "A multi-objective genetic algorithm based approach for energy efficient QoS-routing in two-tiered wireless sensor networks," in *IEEE International symposium on Wireless Pervasive Computing (ISWPC)*, 2010.

[32] R. Krishnamachari and T. Bhaskar, " Networking wireless sensors," in *Cambridge University Press*, Cambridge , 2005.

[33] P. Ezhumalai, P. Periyathambi, S. K. Manoj and O. Chokkalin, "Tree Based Aggregation Algorithm Design Issues," *Wireless Sensor Network,* vol. 1, no. 4, pp. 55-61, December 2009.

[34] I. Njini and O. O. Ekabua, "Genetic Algorithm Based Energy Efficient Optimization Strategies in Wireless Sensor Networks: A Survey.," *ACSIJ Advances in Computer Science: An International Journal,* vol. 3, no. 5, pp. 1-9, 2014.

[35] Y. S. Yen, Y. K. Chan, H. C. ChaoH and J. H. Park, "A genetic algorithm for energy-efficient based multicast routing on MANETs," *Computer Communications,* vol. 31, no. 4, pp. 858-869, December 2008.

[36] M. Youssef, M. Ibrahim , M. Abdelatif and L. Chen, "Routing metrics of cognitive radio networks , A survey," *IEEE Communications Surveys & Tutors,* vol. 16, no. 1, pp. 92-109, 2014 Jan 1.

[37] A. I. Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE wireless communications,* vol. 11, no. 6, pp. 6-28, 2004 .

[38] A. Nasipuri, Castañeda and S. R. Das, "Performance of multipath routing for on-demand protocols," *Mobile Networks and applications,* vol. 6, no. 4, pp. 339-349, 2001.

[39] A. Boukerche, X. Cheng and J. Linus, "A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks," *Wireless Networks,* vol. 11, no. 5, pp. 619-635, 2005.

[40] X. Li, F. Bian, R. Govidan and W. Hong, "Rebalancing distributed data storage in sensor networks," http://www. cs. usc. edu/Research/TechReports/05-852. pdf., 2005.

[41] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," *Global Telecommunications Conference, GLOBECOM'03. IEEE,* vol. 1, no. 3, pp. 548-552, 2003.

[42] N. Baccour, A. Koubâa, M. B. Jamâa and H. Youssef, " A comparative simulation study of link quality estimators in wireless sensor networks," *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems,* vol. 12, no. 6, pp. 1-10, 2009.

[43] A. Gupta, M. Sharma, M. Marot and M. Becker, "Hybrid multihoplqi for improving asymmetric links in wireless sensor networks," in *GuHybridlqi: Hybrid Telecommunications (AICT), Sixth Advanced International Conference. IEEE.*, 2010.

[44] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless networks," in *19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[45] J. He, S. Ji, M. Yan, Y. Pan and Y. Li, " Load–balanced CDS construction in wireless sensor networks via genetic algorithm," *International Journal of Sensor Networks,* vol. 11, no. 3, pp. 166-178, 2012.

[46] J. He, S. Ji, P. Fan, Y. Pan and Y. Li, "Constructing a load-balanced virtual backbone in wireless sensor networks," in *iComputing, NeInternational Conference in Computing, Networking and Communications (ICNC)*, 2012.

[47] T. Back, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford : Oxford university press., 1996.

[48] A. Raha, M. K. Naskar, P. Avishek and C. Chakraborty, "A genetic algorithm inspired load balancing protocol for congestion control in wireless sensor networks using trust based routing framework

(GACCTR)," *International Journal of Computer Network and Information Security,* vol. 5, no. 9, p. 9, 2013.

[49] P. M. Pradhan and G. Panda, "Pareto optimization of cognitive radio parameters using multiobjective evolutionary algorithms and fuzzy decision making," *Swarm and Evolutionary Computation,* vol. 7, no. 1, pp. 7-20, 2012.

[50] S. K. Singh, M. P. Singh and D. K. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks," *Journal of Advanced Networking and Application (IJANA),* vol. 2, no. 2, pp. 570-580, Feburuary 2010.

[51] T. Banerjee, B. Xie and D. P. Agrawal, " Fault tolerant multiple event detection in a wireless sensor network," *Banerjee, T., Xie, B., & Agrawal, D. P. (2008). Fault tolerant multiple event deteJournal of parallel and distributed computing,* vol. 68, no. 9, pp. 1222-1234, March 2008.

[52] E. Felemban, C. G. Lee and E. Ekici, "MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks," *IEEE transactions on mobile computing,* vol. 5, no. 6, pp. 738-754, 2006.

[53] L. Guo and Q. Tang, " An improved routing protocol in WSN with hybrid genetic algorithm," in *Second International Conference onIn Networks Security Wireless Communications and Trusted,* 2010.

[54] E. Alotaibi and b. Mukherjee, " A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer networks,* vol. 56, no. 2, pp. 940-965., 2012.

[55] S. Yang, C. K. Yeo and B. S. Lee, " Toward reliable data delivery for highly dynamic mobile ad hoc networks," *IEEE transactions on mobile computing,* vol. 11, no. 1, pp. 111-124, 2012.

[56] K. Zeng, W. Lou, J. Yang and D. R. Brown III, "On throughput efficiency of geographic opportunistic routing in multihop wireless networks," *Mobile Networks and Applications,* vol. 12, no. 6, pp. 347-357, 2007.

[57] L. Cheng, J. Niu, J. Cao, S. K. Das and Y. Gu, "QoS aware geographic opportunistic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, no. 7, pp. 1864-1875, 2014.

[58] C. S. Raghavendra, K. Viktor and D. Prasanna, "Distributed Signal Processing in Wireless Sensor Networks," *UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES DEPT OF ELECTRICAL ENGINEERING,* pp. 12-35, 2005.

[59] F. Sheng, " Exploring DCV Side Information Based on Various Schemes," *Recent Patents on Computer Science, 7(2), 75-109.,* vol. 7, no. 2, pp. 75-109, December 2014.

[60] P. Wang, C. Li and J. Zheng, "Distributed data aggregation using clustered slepian-wolf coding in wireless sensor networks," *IEEE International Conference in Communicatins,* vol. 8, no. 2, pp. 3616-3622, June 2007.

[61] S. Pattem, B. Krishnamachari and R. Govindan, " The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 4, no. 4, pp. 213-234, October 2008.

[62] Z. Xiong, A. D. Liveris and S. Cheng, " Distributed source coding for sensor networks," *IEEE Signal Processing Magazine,* vol. 21, no. 5, pp. 80-94, January 2004.

[63] Hanzo, L. Lajos L, C. Peter and S. Jurgen , Video compression and communications: from basics to MPEG4 for DVB and HSDPA-style adaptive turbo-transceivers, H. John Wiley & Sons., Video compression and communications.

[64] M. C. Vuran, Ö. B. Akan and I. F. Akyildiz, "Vuran, M. C., Akan, Ö. B., &Spatio-temporal correlation: theory and applications for wireless sensor networks," *Spatio-tempoComputer Networks,* vol. 45, no. 3, pp. 245-259, March 2004.

[65] B. Alam, G. Wang and J. Cao, "Energy and bandwidth-efficient wireless sensor networks for monitoring high-frequency events," in *10th Annual IEEE Communications and Networks*, 2013.

[66] W. Wang, D. Peng, H. Wang, H. Sharif and , "Wang, W. Energy efficient multirate interaction in distributed source coding and wireless sensor network," *IEEE in Communications and Neworking,* vol. 24, no. 5, pp. 4091-4095, March 2007.

[67] J. Haupt , R. Nowak and R. Castro, "Hau Adaptive sensing for sparse signal recovery," *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop,* vol. 13, no. 2, pp. 702-707, January 2009.

[68] W. W. Dargie and C. Poellabauer, Fundamentals of wireless sensor networks: theory and practice., John Wiley & Sons., 2010.

[69] S. J. Wei, X. L. Zhang, J. Shi and G. Xiang, "Sparse reconstruction for SAR imaging based on compressed sensing," *Electromagnetics Research,* vol. 6, no. 3, pp. 63-81, March 2010.

[70] F. Krahmer and R. Ward, "New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property," *Journal on Mathematical Analysis,* vol. 43, no. 3, pp. 1269-1291, March 2011.

[71] E. J. Candes, M. B. Wakin and S. P. Boyd, " Enhancing sparsity by reweighted ℓ 1 minimization," *Candes, E. J., Wakin, M. B., & Boyd, S. P. (2008). Enhancing spa Journal of Fourier analysis and applications,* vol. 14, no. 5, pp. 877-905, December 2008.

[72] W. Yin, S. Osher, D. Goldfarb and J. Darbon, " Bregman iterative algorithms for \ell_1-minimization with applications to compressed sensing," *Journal on Imaging Sciences,* vol. 1, no. 1, pp. 143-168, 2008.

[73] K. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization,* vol. 6, no. 15, pp. 615-640, March 2010.

[74] K. Papafitsoros and C. B. Schönlieb, " A combined first and second order variational approach for image reconstruction," *Journal of mathematical imaging and vision,* vol. 48, no. 2, pp. 308-338, July 2014.

[75] M. F. Duarte, S. Sarvotham, D. Baron and M. Wakin, " Distributed compressed sensing of jointly sparse signals," in *Conference in Signals and Systems Computing*, Asilomar, 2005.

[76] P. R. Grassi, V. Rana, I. Beretta and D. Sciuto, "Grassi, P. R., Rana, V., Tacit Consent: A Technique to Reduce Redundant Transmissions from Spatially Correlated Nodes in Wireless Sensor Networks," in *15th Euromicro Conference in Digital Systems Design*, 2012.

[77] H. Hematkhah and Y. S. Kavian, "Distributed Clustering Protocol Using Voting and Priority for Wireless Sensor Networks," *Sensors,* vol. 15, no. 3, pp. 5763-5782, March 2012.

[78] T. Ahmet and N. S. Erdogan, "Design of Reconfiguarable FFT Processor using Multi-Objective genetic algorithm," *First NASA/ESA Conference on Adaptive Hardware and Systems,* vol. 13, no. 1, pp. 1-2, september 2006.

[79] A. T. Erdogan, M. W. Hasan and H. T. Arslan, "Low Power Commutator for Pipelined FFT Processors," *IEEE,* vol. 19, no. 7, pp. 1-3, May 2005.

[80] C. F. Chiasserini, "On the concept of distributed Digital signal processing in wireless sensor networks," in *Proceeding of MILCOM, 2002 , pp. 260-264*, 2002.

[81] T. Canli and A. K. Gupta, "Power efficient algorithm for computing Fast Fourier Transform over wireless sensor networks," in *IEEE*, Chicago, 2006.

[82] T. Canli, A. Khokhar, M. Terwilliger and A. Gupta, "T. Canli, A. Khokhar. " Terwilliger, M. A. Gupta, Power-Time Efficient Algorithm for Computing FFT in sensor networks," *IEEE,* vol. 14, no. 6, pp. 87-90, December 2003.

[83] T. Arslan, A. T. Erdogan and M. Hasan, "Low Power Commutator for Pipelined FFT Processors," in *University of conference on Low Power Commutator for Pipelined FFT Processors*, Edinburgh, July.

[84] W. Zhao, T. Shuo, T. Xin and L. A. Ming, "Implement of Evolable Hardware based Improved Genetic Algorithm," in *Seventh International Conference on Natural Computation*, AlgoShaanxi, 2011.

[85] K. Boopatthy and B. D. Dhanasekaran, "Fault Tolerant Dynamic Antenna Array in Smart Antenna System using Evolved Virtual Reconfigurable Circuit," in *21st International Conference of VLSI Design*, 2008.

[86] H. Kyler, "Evolvable hardware in Theory and Implementation," in *UMM CSci Seminar conference* , Morris, 2010.

[87] D. Dhanasekaran and K. B. Bagan, "Fault Tolerant Dynamic Antenna Array in Smart Antenna System Using Evolved Virtual Reconfigurable Circuit," in *IEEE, 21st International Conference on VLSI Design* , 2008.

[88] W. Vanderbauwhede and K. Benkrid, High-performance computing using FPGAs, New York: Springer., 2013.

[89] C. Zhu, C. Zheng, L. Shu and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications,* vol. 35, no. 2, pp. 619-632, 2012.

[90] B. Seshasayee, "Middleware-based services for virtual cooperative mobile platforms," *ProQuest,* vol. 4, no. 2, pp. 44-49, December 2008.

[91] W. N. Schilit, " A system architecture for context-aware mobile computing," Columbia , 1995.

[92] A. K. Dey, G. D. Abowd and D. Salber, " A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-computer interaction,* vol. 16, no. 2, pp. 97-166, July 2001.

[93] K. S. Phanse, A. Bhat and L. A. DaSilva, "Modeling and evaluation of a policy provisioning architecture for mobile ad-hoc networks," *Journal of Network and Systems Management,* vol. 14, no. 2, pp. 261-278, September 2006.

[94] E. E. Egboah, "A survey of system architecture requirements for health based wireless sensor networks," *A survey of Sensors,* vol. 11, no. 5, pp. 4875-4899, December 2011.

[95] Y. Jia, T. Dong and J. Shi, "Analysis on energy cost for wireless sensor networks," *IEEE International Conference in Embedded Software and Systems,* vol. 2, no. 2, pp. 222-234, December 2005.

[96] J. Han, W. Zhao and M. Zheng, "An analytical model on unbalanced energy consumption in large scale wireless sensor network," in *Third International Conference Technology and Mechatronic animation (ICMTMA)*, 2011.

[97] P. Hu, Z. Zhou, Q. Liu Q and F. Li, "The HMM-based modeling for the energy level prediction in wireless sensor networks," *IEEE conference on Industrial Electronics and Applications(ICIEA),* vol. 4, no. 2, pp. 2253-2258, May 2007.

[98] J. Hill, R. Szewczyk, A. Woo and S. C. Hollar, "System architecture directions for networked sensors," in *ACM International conference on operating systems review*, 2004.

[99] G. J. García, C. A. Jara, J. Pomares and A. Alabdo, "García, G. J., Jara, C. A., Pomares, J., A survey on FPGA-Based sensor systems: towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing," *Sensors,* vol. 14, no. 4, p. 624, August 2014.

[100] F. W. Liou, Rapid prototyping and engineering applications: a toolbox for prototype development, CRC Press., Liou, F. W. (2007). Rapid prototyping and engineering applications: a toolbox for prototype development. CRC Press..

[101] P. Jamieson, T. Becker, W. Luk and P. Y. Cheung, " Benchmarking reconfigurable architectures in the mobile domain," *IEEE ,* vol. 8, no. 3, pp. 131-138, Aprial 2009.

[102] A. Conti, D. Dardari and R. Verdone, "A Collaborative Signal processing for energy –efficient self-organising wireless sensor network," in *International workshop on wireless AD-hoc Network IEEE,* 2004.

[103] P. Pati, R. Prakashgoud R, P. Umakant and E. Kulkarn, "Energy-Efficient Cluster-Based Aggregation Protocol for Heterogeneous Wireless Sensor Networks," in *HeterogIntelligent Computing, Networking, and Informatics*, India, 2014..

[104] G. EkbataniFard and M. Reza , "Fast Multi-objective Genetic Algorithm based Approach for Energy Efficient QoS-Routing in Two-tiered Wireless Multimedia Sensor Networks," *Modern Applied Science,* vol. 12, no. 6, pp. 35-41, November 2010.

[105] X. Zhang, M. Burger, X. Bresson and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *Journal on Imaging Sciences,* vol. 3, no. 3, pp. 253-276, December 2010.