CRANFIELD UNIVERSITY

Moner Alsader

Power Management for Energy Harvesting

School of Aerospace, Transport and Manufacturing Aerospace

PhD

Academic Year: 2014 - 2020

Supervisor: Dr Ivan Petrunin and Professor Antonios Tsourdos

May 2020

CRANFIELD UNIVERSITY

School of Aerospace, Transport and Manufacturing Aerospace

PhD

Academic Year: 2014 - 2020

Moner Alsader

Power Management for Energy Harvesting

Supervisor: Dr Ivan Petrunin and Professor Antonios Tsourdos

May 2020

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD

ABSTRACT

The use of wireless sensor networks in aircraft health management grew exponentially over the past few decades. Wireless sensor networks provide technology that reduces the amount of wiring for aircraft, thereby reducing the weight and cost of aircraft. One of the most significant limitations in the use of wireless sensor networks in aircraft health management systems is the availability of power sources. Developing Wireless Sensor Network nodes that can generate and harvest their autonomous power supply continuously is a bottleneck that has been the preoccupation of engineers for many years.

The amount of energy a network of Wireless Sensors can harvest fluctuates and is difficult to predict. As a result, existing predictors of energy harvesting are prone to errors. Models-free schemes such as expert systems are thus preferred for energy management strategies.

The main aim of this thesis is to propose expert-based systems for energy harvesting in aircraft to enhance wireless sensor nodes life span by improving energy harvesting, energy storage and packet loss probability.

In this context, a novel integrated approach based on the Markov chain was proposed for energy harvesting in aircraft. Simulation results and quantitative analysis showed that the integration of Piezoelectric and Thermoelectric harvesters with stochastic scheduling had a better performance in terms of energy storage, energy harvesting and packet loss probability. There was also an increase in energy storage with five Markov states compared to that of two Markov states. The packet loss probability of the integrated approach with five Markov states was better than that of two Markov states. The results also showed that the integrated approach with five Markov states harvested more energy than two Markov states.

The novel integration of LTspice and NS-3 simulators was proposed. The LTspice and NS-3 integration was validated by deploying the Fuzzy logic control approach in energy harvesting. Simulation results and quantitative analysis based on Fuzzy control logic expert system indicated that the in-

tegration of LTspice and NS-3 was found to be better in energy harvesting compared to non-fuzzy control systems. The downtime ratio and energy utilization efficiency of the wireless sensor nodes were also found to be better than non-fuzzy control.

The power management based LEACH routing protocol was also proposed. The simulation results and quantitative analysis showed that the average harvested energy based on the LEACH routing protocol deployed with fuzzy logic and Markov chain was better compared to those with direct communication based on Markov chain and fuzzy logic systems.

Keywords:
Piezoelectric generators, Thermoelectric generators, Fuzzy, Markov models, LEACH routing protocol, NS-3, LTspice

ACKNOWLEDGEMENTS

I would like to express my thanks for the help and guidance that my supervisors Dr Ivan Petrunin and Professor Antonios Tsourdos, have given me. They gave guidance I needed during my study. Their experience and expertise were beneficial, and under challenging conditions, they helped me to complete my research.

I am thankful to all my university colleagues. They genuinely contribute to my own and academic growth.

I do want to thank my family for the support they have provided. My family gave me time to finish my research. I want to say how grateful I am to my wife for her patience and understanding during my studies. She had been very helpful.

PUBLICATIONS

Moner Alsader and Al Savvaris. Integrated LTspice and NS-3 Power Management Simulation for Energy Harvesting. International Journal of Applied Mathematics and Informatics. pp. 120-125. Vol 1, 2017, ISSN: 2074-1278.

Moner Alsader and Al Savvaris. Integrated Power Management for Energy Harvesters in Aircraft. Proceedings of the 24th International Conference on Automation & Computing, Newcastle University, Newcastle upon Tyne, UK, 6-7 September 2018

Moner Alsader and Al Savvaris. Power Management for Energy Harvesting over LEACH Protocol. International Journal of Computers. pp. 18-22. Vol 13, 2019, ISSN: 1998-4308.

DEDICATION

A late thesis to the memory of my late father.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
ADC	Analog-to-digital Converter
AES	Advanced Encryption Standard
AHM	Aircraft Health Management
AI	Artificial Intelligence
AIR-MAN	Aircraft Maintenance Analysis
BS	Base Station
CDMA	Code Division Multiple Access
CH	Cluster Head
COMAC	Commercial Aircraft Corporation of China
CPU	Central Processing Unit
DC	Direct Current
DD	Directed Diffusion

DEHAR Distributed Energy Harvesting Aware Routing

EAR Energy Aware Routing

FCFS First COME First SERVED

FPGA Field-Programmable Gate Array

GM Generalized Markov

GPSR Global Positioning System Receiver

IC Integrated Circuit

IEEE Institute of Electrical and Electronics Engineers

ISM Industrial, Scientific, and Medical

LEACH Low Energy Adaptive Clustering Hierarchy

LTspice Linear Technology Simulation Program with Integrated Circuit Emphasis

MCU Microcontroller Unit

NS2	Network Simulator 2
NS3	Network Simulator 3
OMNET	Objective Modular Network Testbed
OPNET	Optimized Network Engineering Tools
PAN	Personal Area Network
PCB	Printed Circuit Board
PEG	Piezoelectric Energy Generator
PEH	Piezoelectric Energy Harvesting
PHM	prognostics and health management
PMN-PT	Lead Magnesium Niobate-Lead Titanate
RAM	Random Access Memory
RF	Radio Frequency
RMS	Root Mean Square

SEPIC	Single-ended Primary-Inductor Converter
SHM	Structural Health Monitoring
SPICE	Simulation Program with Integrated Circuit Emphasis
SPIN	Sensor Protocols for Information via Negotiation
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TEEN	Threshold-sensitive Energy-Efficient Sensor Network
TEG	Thermoelectric Energy Generator
UDP	User Datagram Protocol
WSN	Wireless Sensor Network

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Overview

A paradigm change in the data generation and the collection has been achieved by using the Wireless Sensor Network (WSN). In general, WSN consists of a large number of sensor nodes that have individual computing capabilities. It permits the collection of data from sources like the environment and wirelessly transmits data into a server or management/control module. Data can be transmitted and received by the sensor modules. A major attraction of the WSNs is that due to their small size and low cost, several of them can deployed. As a result, WSNs have found application in many varied and diverse applications such as,

- Military: In military communication, control, intelligence and surveillance, WSNs play a critical role. WSN may be used to continuously detect and track enemies' vehicles [3].
- Environment: The WSNs have a significant role in earthquakes, coal mining, tsunamis, flood warning, gas leakage, forest fire prediction, cyclones. This helps to measure harmful air pollutants and primary weather particles. WSNs are also used for earlier detection of air pollutants [4].
- Agriculture: The WSNs are used to detect parameters such as temperature and pressure in precision agriculture and also ensure a perfect climate for crop growing [5].
- Vehicle Tracking: The WSNs help to avoid traffic congestion as well as they help to manage car parking systems and vehicles' location [6, 7, 8].
- Animal Tracking: Vibration and movement can be monitored by using the WSNs for the animals. The WSNs can also be used for optimising the rearing conditions and controlling the stress level of the animals [9, 10].

- Health Monitoring: WSNs play a crucial role in physiological signals monitoring in real-time and avoid danger to life [11, 12].

WSNs also find application in the aerospace industry. Modern aircraft systems involve multidisciplinary engineering such as aeronautical, electrical and electronics, computer science, mechanical and electromechanical and Radio Frequency. It is becoming challenging to manage new aircraft systems. Commercial aircraft fly thousands of miles daily with restless departures and landing schedules. With combat aircraft, everything is the same because they are sometimes subjected to rigorous war operations and battle operational exercises. Human lives are at risk in all these situations if the maintenance of the aircraft is not planned on time. Unless the maintenance and health monitoring programme is given some consideration, maintenance breakdowns are often required. The cost and time would be required as additional resources for breakdown repairs. Real-time safety tracking of these complex systems can be carried out with developments in WSN technology. WSNs can continuously monitor the health of the different components of the aircraft systems and preventative measures needed before breakdowns [13].

A WSN, with sensors integrated with energy harvesters, are designed and adapted for Structural Health Monitoring (SHM), wherein continuous integrity of the structure can be monitored to assess maintenance or necessary course of action for the safe operation of the structure. The continued operation of these sensor networks is achievable through the energy harvesters that can be integrated with the sensor nodes to meet their low energy requirements. The energy is directed to the microprocessor, Radio Frequency (RF) transceiver, rechargeable battery or supercapacitor from energy harvesting circuit, wherein energy arrives in intermittent levels in the form of the thermal, kinetic and vibrations. Communication through RF transceiver is subjected to constraints at multiple levels in the architecture of the sensor node. These communication constraints could be roughly categorized into energy, memory, computational speed and communication bandwidth.

1.2 Problem Statement

Individual aircraft systems have gradually increased in number and complexity to such a level that it has become necessary to develop systems that can automatically monitor the health of onboard systems, and were possibly solved potential problems. Aircraft Health Management (AHM) systems are highly desirable in the aerospace industry. These systems are used as fly-by-wire technologies for routine monitoring and diagnosis of aircraft systems in modern aircraft. Some of the leading aircraft manufacturers have developed propriety AHMs. For example, Boeings' aeroplane health management system, Aircraft Maintenance Analysis (AIR-MAN) system and prognostics and health management (PHM) developed by the Commercial Aircraft Corporation of China (COMAC). The common objective of these systems is to remotely collect data on all the systems or modules of an aircraft, characterise the health status of the systems based on intelligent analysis of the collected data and thereby determine the overall health status of the aircraft and to determine service scheduling times.

The use of WSNs in AHMs has exponentially increased over the past few decades [14, 15, 16]. This is because WSNs provide an enabling technology to reduce the amount of aircraft cabling, thereby reducing aircraft weight and cost and decreases carbon emissions. Wired-based aircraft sensor systems are heavy and vulnerable to wear damage and decay. The Airbus A380, consisting of about 98,000 wires and 40,000 connections, and carries over 300 miles of cables. [17, 18]. Featuring nearly 10,000 sensors on every single wing, the A380-1000 is the super-sized airliner for transporting up to 1,000 passengers [19].

Replacing wired sensors with wireless sensors will accomplish the objective of increasing the number of sensors consequently reducing the weight of the aircraft system. This would also increase fuel efficiencies and in return, minimise carbon emissions [20].

However, one of the main limitations of the use of WSN's in AHM systems is the limited power source of WSN nodes. WSN nodes are powered from the main by batteries which can sometimes be large, have low voltage capacities and need to be either replaced or recharged hence increasing the

costs involved [21].

Developing WSN nodes that can generate and harvest their autonomous power supply continuously is a bottleneck that has been the preoccupation of engineers for many years. A WSN-based AHM in which energy is autonomously and continuously harvested is highly desirable. Harvesting energy to power WSN has become feasible with advances in microelectronics, sensors and microcontrollers that have low power requirements, and WSN nodes that have wireless transceivers with low power requirements. The harvesting of energy based on such technologies has been the focus of researchers and designers because they can be used to remove the need for battery-operated WSNs which will increase their lifespan and reduce costs involved in WSN-based AHMs.

WSN is important for SHM in aircraft as it consists of continuous monitoring of key parameters for the ageing effects prediction and its impact. The main goal is to minimise costs and enhance mechanical health through predictive maintenance. WSN nodes are needed for efficient SHM to have sufficient energy to pass critical parameters for monitoring purposes. The harvesting of energy from an ambient source, like vibration and heat, is necessary to power WSN nodes to ensure enough energy is available at critical times of data monitoring and transmission.

Energy harvesting and management in WSNs is, therefore, very topical and an active research area. There are different vibration sources and thermal gradient that are present in an aircraft that are potential energy sources that can be harvested. The energy sources within an aircraft and how these can be harvested for aerospace application purposes are detailed in Le et al., [22]. A demonstration of energy harvesting from aircraft is discussed and demonstrated in [23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. The majority of energy harvesting systems for AHMs are based, however, on the technology of thermoelectric energy harvesting.

The energy quantity a WSN can harvest fluctuates and is unpredictable. Energy harvesters prediction models are therefore prone to errors, either they overuse or underuse the quantity of the energy harvested. For power management strategies, prediction model-free schemes such as the expert

system are thus preferred.

We hypothesise that it may be possible to design an expert system WSN-based AHM that is powered by energy that is harvested from vibration energy from the aircraft. In this study, we design and simulate an expert system based power management system for aircraft health monitoring using wireless sensor networks.

1.3 Aims and Objectives

Due to a possible limitation of harvested power, the proposed power management system should be energy efficient whereby the harvested energy is only consumed by the sensor nodes when necessary. A critical requirement is the design of an intelligent power management system that will optimise the use of power and ideally conserve energy. The thesis proposes the use of expert systems, fuzzy rule-based systems, in the design of the power management system.

Therefore, the specific aim of this thesis is to propose a novel expert based WSN power management system for effective utilisation of energy harvested from vibration and thermal sources in an aircraft health monitoring system.

The following objectives are set to fulfil the aim of this study,

1. Conduct literature review on state of the art in vibration and thermal energy harvesters for WSN systems in aircraft AHM system and energy-efficient communication and routing protocols that can be deployed to minimize power consumption in aircraft for AHM systems.
2. Formulate a Markov model and fuzzy expert system for node-level decisions for battery health, recharging urgency and priority, and data transmission.
3. Develop and simulate Markov and fuzzy expert systems by using Linear Technology Simulation Program with Integrated Circuit Emphasis (LTspice) and Network Simulator 3 (NS3) simulation tools.

4. Develop and simulate energy-efficient communication and routing protocols for AHM.

5. Conduct comparative analysis between Markov prediction models and fuzzy expert systems for power management systems in aircraft for AHM.

1.4 Methodology

Various tasks were carried out as part of this thesis. Firstly, a review of the literature was carried out to determine the state-of-the-art in WSN-based AHM systems where the energy is harvested through ambient sources such as vibration and thermal. Various ambient sources were reviewed, which includes, solar, vibration, thermoelectric, wind flow, acoustic noise and electromagnetic. It was concluded that vibration and thermoelectric ambient sources were appropriate in AHM systems. Vibration and thermal energy are in abundance in the environment of an aircraft, and this energy can be harvested on-board power sensor. Although solar is very useful and widely used, this source of energy harvesting has a limitation, and it cannot be used for applications and locations where access to the light source cannot be guaranteed.

The second stage is the design of the system itself. The design was based on Markov chain models and expert system based on a fuzzy logic system. These models were to give decisions for battery health, recharging urgency and priority and data transmission. The set of actions were based on activities such as recharging, sleep mode and data transmission. Several computational intelligence techniques such as neural network and genetic algorithms, statistical, probabilistic, machine learning and fuzzy control methods and their applications in WSN were reviewed. It was found that the expert system, based on a fuzzy logic approach, is capable of practical enhancing vibration power harvesters in WSN while other systems which are based on prediction models were shown to be prone to errors, either over-use or under-use of the harvested energy.

The third stage involved developing and simulating the designed Markov

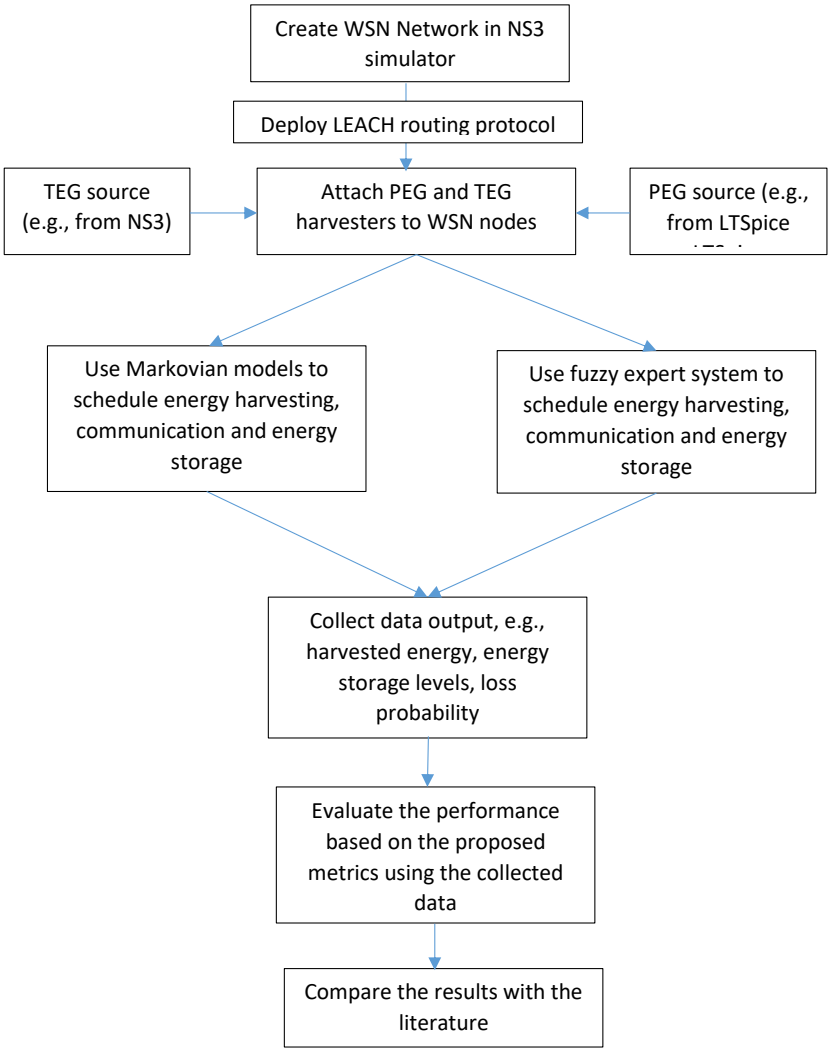

Figure 1.1. Methodology stages

models and expert-based systems. The development and simulation were based on C++ programming using the NS3 simulation tool. The LTspice tool was also used to supply harvested energy to the NS3. Several networking simulation tools such as NS3, Network Simulator 2 (NS2), Objective Modular Network Testbed (OMNET) and Optimized Network Engineering Tools (OPNET) were reviewed with their applications in WSN. NS3 was found to be up to date with several networking protocols, wireless sensors and energy harvesting models. Circuit design simulation tools such as LTspice and Simulink were also reviewed, LTspice was found to have up to date energy harvester tools. LTspice and NS3 are used in this thesis because they are widely used by the research community for both communication and circuit simulation purposes. The final stage of the thesis is the testing of the sim-

ulation system to determine if it met the requirements, design specifications and objectives set at the beginning of this research.

These stages are depicted in Figure 1.1.

1.5 Novel Research Contributions

1. Novel integration of vibration and thermal energy harvesters to mitigate the uncertainty of reliable energy consumption in WSN nodes is proposed in this thesis. The integration is based on the Markov chain model for scheduling scheme to minimize packet loss probability and enhance efficient communication in WSN nodes. The novelty of this contribution is based on the thermal and vibration integration in WSN nodes.

2. Novel expert system based power management for energy harvesting in an aircraft health monitoring system is proposed and investigated. The expert system, based on a fuzzy logic approach, is capable of practical enhancing of vibration power harvesters in wireless sensor nodes. Current systems are based on prediction models, which have shown to be prone to errors, either overuse or underuse of the harvested energy. The novelty of this contribution is based on the development of the fuzzy logic system for enhancing energy harvesting in WSN nodes.

3. An integrated simulation model for energy harvesting WSN applications based on the combination of LTspice and NS3 is proposed and implemented for simulation of AHM WSN application. The current popular simulation tools for electronics (LTspice) and networking (NS3) are being used in isolation. Integrated approach in this thesis proposes to use these two simulation tools in conjunction through virtualization, communication channel (Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)) between LTspice and NS3. LTspice parameters, such as harvested power can be communicated to the NS3 simulation in real-time. The novelty of this contribution is based on the integration of two popular simulators in electronics and networking domains (LTspice and NS3)

4. WSN for AHM utilising Low Energy Adaptive Clustering Hierarchy (LEACH) communication protocol is proposed and simulated. The existing power management systems have been deploying routing protocols that contribute to high energy consumption. The proposed approach is proven to be efficiently conserving more energy and hence prolonging the lifetime of the wireless sensor nodes. The novelty of this contribution is based on the deployment of energy-saving LEACH routing in the proposed approaches.

1.6 Research Impact

The results obtained in this research will be beneficial for aerospace stakeholders such as Airbus and Boeing to optimise the use of power and conserve energy in their aircraft structural health monitoring management systems. It uses real-time data output from wireless sensors on different parts of the aircraft to enhance the reliability and safety of the aircraft. The energy conserved by the proposed expert system will help to prolong the lifetime of the wireless sensor nodes and hence provide reliable data transmission uninterruptedly throughout an aircraft flight. The introduction of the expert system and implemented the proposed integrated power management for energy harvesters were able to increase the residual energy by up to 60% compared to individual harvesters. Although the expert system performs better than that of stochastic based system, through empirical investigation, the processing time increased to at most 10%. This was because the inference engine had to process a significant number of rules to reach the required decisions.

The proposed energy-efficient routing protocol can be used to provide fast and reliable data transmission for aircraft health monitoring systems compared to WSN solutions utilising direct communication mode. This is because fast and reliable communication is a lifeline to the aircraft health monitoring systems. In this study, it was demonstrated that with LEACH procol, there was an improvement of about 62% over direct communication model in terms of residual energy. The number of dead nodes was also significantly reduced by 40% compared to the direct communication mode.

The developed simulations and the dataset obtained as the result of the system development and simulation can benefit the research community. This is because the data obtained in the study can be shared and used by other researchers in academia and industry to perform comparative analysis or use them to further enhance their system model.

The use of WSN eliminates wired networks. This will in return reduces the weight of the aircraft and hence, reduce aircraft fuel consumption. The reduction of aircraft fuel consumption will reduce co_2 emission. Aviation is responsible for 12% of co2 emission of total transport [20].

1.7 Structure of the Thesis

The thesis is structured as follows. The background aims, objectives and contribution of the thesis are presented in **Chapter 1**. Review of the relevant literature such as technologies, energy harvesting technologies, wireless network standards, power management systems and fuzzy control systems is presented in **Chapter 2** which also identifies gaps in the literature. These gaps are in the power management system for energy harvesting from vibration and thermal sources in AHM, the deployment of energy-efficient routing protocols and deployment of simulation tools. This chapter fulfils the **objective 1**.

One of the gaps identified in **Chapter 2** will be dealt with in **Chapter 3** where the integrated power management for energy harvesters will be proposed, this approach integrates thermal, and vibration sources, they are controlled and managed by the Markov chain to maximize harvested energy and storage as well as to minimize data loss caused by insufficient harvested energy in WSNs. This chapter addresses **objectives number 2 and 5 and contribution number 1**.

Chapter 4 fills the gap in simulation tools and power management systems by integrating LTspice and NS3 to simulate power management for energy harvesting. This approach uses a fuzzy control system to improve downtime ratio and energy efficiency. This chapter fulfils **objectives number 3 and 5**

and contributions number 2 and 3.

The gap in energy-saving routing protocols deployment is addressed in **Chapter 5** where power management for energy harvesting over LEACH routing protocol is presented. This chapter addresses **objectives number 4 and 5 and contribution number 4**. The conclusions, strengths and weaknesses of this study and potential future extension of the work are discussed in **Chapter 6**.

2 LITERATURE REVIEW

The background technologies that inspire the study towards this thesis are discussed in this chapter. The chapter starts with the detailed energy harvesting technologies which include energy harvesting sources such as solar, wind, vibrations, thermoelectric, acoustic and magnetic. Energy storage and wireless networks standards such as Zigbee and Bluetooth are discussed. Wireless sensor architecture is presented in detail together with LEACH routing protocol. Power management and energy optimization techniques together with optimization tools, which are the core area of this thesis are discussed in detail. The chapter concludes with a summary.

2.1 Energy Harvesting Technologies

Energy harvesting is a method used to collect and extract surplus ambient energy, such as wind, vibrational, solar and thermal and to transform the harvested energy into the stored electrical energy, ready to be used for sensing or actuating purposes. In contrast to large-scale energy harvesting using renewable energy sources such as wind and solar farms, the harvested energy is typically minimal. In comparison to large energy harvesting stations at a given location, small energy sources are compact and readily available for use. Energy from the ambience is used to power small independent sensors, which are used for remote sensing purposes while they are exposed to hostile conditions in the long term. The functions of these small independent sensors are often constrained by battery energy dependence. Therefore, the driving force for energy harvesting technologies is that of supplementing energy storage components such as batteries and capacitors to power wireless sensor networks and mobile devices for extended operations.

Energy harvesting offers the end-users with various benefits, and the following list contains some of the main advantages in terms of the energy harvesting appropriateness for WSN. Solutions for energy harvest may:

1. Reduce battery power dependency. The development in microelectronics technology reduces the energy consumption of the sensor nodes and can, therefore, be sufficient to remove the battery from the harvested ambient resources.

2. Minimize the cost of deployment. Self-driven wireless sensor nodes don't need cables and wires, so installation is straightforward, and the cost of the installation is lowered as well.

3. Lower the cost of repair. Energy harvesting helps sensor nodes to work spontaneously after deployment and reduces battery maintenance service visits.

4. Provide continuous sensing and actuation ability in harsh environments.

5. Provide solutions on a long-term basis. So long as ambient energy is available, a robust auto-powered sensor node can remain operational.

6. Autonomous sensor nodes are suitable for long-term applications that look at monitoring for decades.

7. Reduces the impact on the environment. The need for millions on battery and battery replacement energy expenses can be eliminated through energy harvesting.

The most mature source of energy harvesting is the use of solar cells which can harvest energy from the light source. The energy thus harvested is generally used to power the outdoor applications. Although very useful and widely used, this source of energy harvesting has a severe limitation, and it cannot be used for applications and locations where access to the light source cannot be guaranteed. For example, this type of energy harvesting may not be very useful for applications inside an aircraft. For this case, other sources of energy that can be harvested may be advantageous. One of the sources of energy from an aircraft that can be harvested is vibration energy. Vibration and thermal energy are in large quantity in the environment of an aircraft [31]. This energy can be harvested to power the on-board systems in the aircraft. We will review applications that harvest power from these energy sources in the following sections.

The power consumption requirement of WSNs must be fully understood before any attempt is made to determine the source of the power to be harvested and what technology should be used for the harvesting. Figure 2.2 shows the characterisation of the power requirement for small electronic devices, including WSNs [33]. It can be seen that the power requirement of a WSN system varies from a few microwatts when in the sleep mode to tens of mill watts when the node is transmitting data. This, therefore, sets the benchmark in terms of power requirements and any power to be harvested should satisfy this power requirement threshold. Table 2.2 shows the amount of power that can be harvested from ambient sources such as light, vibration, thermal and radio frequency.

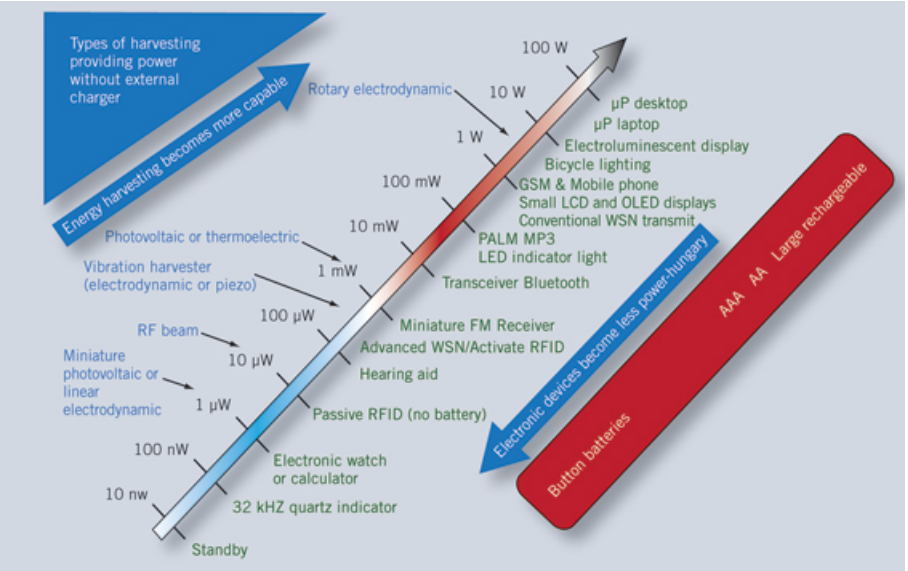

Figure 2.2. Power requirements of small electronic devices [1]

2.1.1 Energy Harvesting Sources

There are many promising sources of energy that are frequently harvested to power WSNs. Table 2.2 lists the most frequently harvested sources of energy that are harvested and the power that can be derived from the harvesting. The most common of these sources is ambient light which can be indoor or outdoor light sources, vibration or motion that can be sourced from humans or industrial plants, thermal which can also be sourced from humans and industrial systems and RF mainly from cellular phones.

Table 2.2. Comparison of power outputs from energy harvesting technologies

Harvesting method	**Power density**
Outdoors solar energy	15mW/cm^3 for a sunny day 0.15mW/cm^3 for a cloudy day [34]
Indoors solar energy	10-100 μW/cm^3 [34]
Shoe inserts vibrations	330 μW/cm^3 [35]
Electrostatic conversions vibrations	0.021 μW/mm^3 – 105Hz 184 μW/cm^3 – 10 Hz [36]
Electromagnetic conversion vibrations	306 μW/cm^3 – 52 Hz [36]
Thermoelectric 5 C gradient	40 μW/cm^3 [37]
Wind flow	16.2 μW/cm^3 – 5m/s [38, 39]
Acoustic noise	3 nW/cm^3 – 75 dB 960 nW/cm^3 – 100 dB [40, 41]
Magnetic field energy	130 μW/cm^3 – 200 μT, 60Hz [42, 43]

The potential power that could be obtained from harvesting these energy sources is shown in Table 2.2 [32]. The power that can potentially be harvested from these energy sources is enough to power WSN-based systems such as WSN-based aircraft health monitoring systems. This opens up a lot of potential for innovation in this domain. Some of the WSNs sensor nodes include radio standard, typical range, data rate, sleep mode, processing consumption, transmission and reception powers, supply voltage and average power consumption per each node. The next subsection discusses the technology used to harvest power from these energy sources. A literature review will be conducted to show the latest work that has been done in harvesting energy from these sources to power WSN systems.

2.1.2 Vibration Sources

For several years, the piezoelectric effect has been used to convert mechanical to electrical energy [44, 45, 46]. The concept behind this is over the two-centuries-old project, although the Curie brothers only demonstrated a definitive correlation between stress and voltage [47]. By then, it has become clear that piezoelectric materials may be electrically polarised or polarised in the event of stress. The stress applied affects the dimensions of a piezoelectric material marginally proportionally, consequently results in a variation in the length of the connections between cations and anions within its internal structure. Conversely, when placed in an electric field, a piezoelectric material has a change in dimension. This reverse process is known as electrostriction [48]. A "poling" method first has to be completed if macroscopic piezoelectricity is to be observed in the material and used for research. It involves exposing the material at a high temperature to a strong Direct Current (DC) electrical field which generally falls below the Curie temperature of the material which forces the material to align and activates the material by leaving an internal polarization [49]. These instruments which convert the mechanical stress into electricity by using the piezoelectric effect are called transducers. They are commonly used in sensing applications such as microphones, sensors and strain gauges. Such equipment using the opposite piezoelectric effect to produce dimensional changes under an applied electric field is referred to as actuators and used as selectors of frequency and positioning devices.

Vibration is one of the sources of energy that is in abundance in aircraft. Harvesting this source of energy require the conversion of energy from one source to another. In this case, it requires converting vibration/motion energy to the electrical energy by using appropriate electromechanical transducers. The most widely used electromechanical transducers that can convert vibration/motion energy into electrical energy are electromagnet, electrostatic, piezoelectric and magnetostrictive. Each of these types of transducers has its advantages and disadvantages outlined in [50]. The type of transducer to use is, therefore, dependent on the application.

The use of piezoelectric transducers has advantages, especially for aircraft-based systems. Piezoelectric transducers produces electrical energy from

kinetic energy that is generated from motion and vibration using the so-called piezoelectric effect. This method is gaining popularity in part due to the increased performance of piezo transducers and technological advances in power Integrated Circuit (IC) such as MEMS sensors and Microcontroller Unit (MCU). Furthermore, these types of transducers are simple, do not generate heat, do not provide unwanted electromagnetic waves that can cause interference and they are simple, therefore easy to manufacture and to use in systems.

2.1.3 Piezoelectric Prototypes

Researchers working at Exeter University were able to develop a high performance WSN based AHM demonstrator in which aircraft wing strain was used as a source of ambient energy. The system harvested energy from vibration, and it was able to generate the power of 2-12 mW at 1-10Hz resonant frequency [51].

Work presented in [52] developed a piezoelectric transducer that was shown to generate electrical power of up to 11.4 mW using vibration energy with a resonant frequency and acceleration of 35.8 Hz and 0.1 g, respectively. Another example of using piezoelectric transducer for energy harvesting is an energy harvester developed by Mide Company. The developed energy harvester was able to generate Root Mean Square (RMS) electrical power of 220 ^{o}W and 440 ^{o}W from vibrations from the aircraft with resonant frequencies of 40Hz and 300Hz, respectively. In [53] a piezoelectric device (Lead Magnesium Niobate-Lead Titanate (PMN-PT)) based energy harvester was able to generate a maximum power of 0.3 mW from a frequency of 1.3 kHz. Work that was reported in [54] was able to generate electric power of up to 1.53 mW with a low resonance frequency of 20.1 Hz.

Work that was presented in [55] reported the design and testing of a piezoelectric based energy harvester that obtained energy conversion efficiency of up to 73%, with harvested electric power of up to 8.4 mW from a vibration source with a resonant frequency of 47 Hz and acceleration of 0.5 g.

A design that was reported in [56] was able to generate up to 240 ℃W of electrical power from a resonant frequency of 66.2 Hz and acceleration of 0.5 g. The results were remarkable given that the size of the energy harvester was only credit card sized. Recently, a novel piezoelectric-based energy harvester was reported in [57]. This system was able to produce up to 100 ℃W of electrical power from vibration energy of 30 Hz resonant frequency and 0.5 g acceleration.

Much work has gone into developing such systems. This is evident from the different ranges of output power and the different types of vibration which is characterised by the resonant frequency. Piezoelectric based energy harvesters are therefore becoming common and are a candidate for a WSN based aircraft health monitoring system – which is the basis of this thesis.

2.1.4 Thermal Sources

Thermal based energy harvesters typically source energy from heat. They, therefore, convert heat to electrical energy using the Seebeck effect. This in effect is based on generating electrical energy from the difference in temperature between two conductors that have different electrical characteristics, thereby generating a voltage difference between the two conductors due to a temperature ifference. The critical components of a harvester based on thermal energy are thermopile which is made up of thermocouples placed between two plates with different temperatures. These generate an output voltage that is proportional to the temperature gradient between the two plates. There has been a lot of research in this area, and research has shown that power in the range of mW can be generated using this method of energy harvesting [32, 50].

Thermoelectric Energy Generator (TEG) is beneficial for the harvesting of energy from the generated temperature during take-off, which can be subsequently harvested during the flight to power systems. The profile of temperature from the skin of an aircraft during a short-haul flight [28] shows that the temperature is subject to changes from 20 ℃ to -22 ℃ within 15 minutes which is followed by a constant temperature of -22 ℃ during cruise flight then

change of temperature from -22 ℃ to 20 ℃ on descent [28]. This is a source of energy that can be harvested and used to power systems autonomously.

TEG has attractive features such as long service life, no moveable parts, quick and high reliability. The low performance is, however, a significant disadvantage which has constantly prevented the wide commercial use of this technology. Only 5-6% of useful heat can be converted into electricity by current TE materials. However, some important studies [58, 59, 60, 61] are being conducted to develop new building materials and modules that offer more than 10 per cent harvest performance.

More recent research has moved into the field of thermal energy harvesting, and successful industrial applications with TEGs have been implemented. The Seiko Thermic watch [62] is the first thermal harvesting application for a consumer product. It uses a thermal energy generator to convert heat from body to electricity used to drive the watch. The generator will harvest 22 µW with a temperature gradient of about 1 K between the wrist and the atmosphere.

Additionally, this energy does not only power the watch, but it also charges a battery of 4.5 mAh lithium-ion. Kocoloski et al., [63] suggest two possible methods to derive electrical energy from waste heat in industrial applications: TEG and thermionic generators. A system was developed to remove waste heat energy from a furnace in a glass production plant using a heating device for testing this idea. The results showed that almost a third of the available energy in exhaust gases could be transformed into electricity.

To generate electrical energy by using waste heat, Eakburanawat [64] developed a TEG microcontroller battery charger. A Single-ended Primary-Inductor Converter (SEPIC) DC-DC-Converter with a maximum charge power of 7.99 W has been used and operated by a microcontroller. The efficiency of the conversion is about 15%.

In [65] the authors suggested a TEG for electricity generation in stoves that could be used for light generation. A generator prototype was built, which should cost about $30 if generated in sufficient quantities. Various design considerations were proposed during their research, for example, selecting

a proper TEG module and designing a heating sink. Also, work is being done to develop a high-efficiency TEG for WSNs.

In [66] the authors built a method for the recovery of thermal energy from the sun. Eight TEG modules were placed in a small greenhouse to enhance its efficiency. When it was put on a 200 DC surface, 40 mW power was generated from such a TEG.

Mateu [67] suggested a thermal generation method for the harvesting of energy from a human body and its atmosphere from the low-temperature gradients. The generation capacity of this TEG was typically as low as 0.3 V, which was not appropriate for many practical applications such as providing power to a wireless communication module and charge a battery. The input voltage increased with a charging pump IC with a step-up DC-DC converter.

A low voltage transformer for TEG was proposed by the authors of [68] whereby, if the input voltage was as low as 300 mV, the transformer would automatically start service. Xin [69] did put forward ZigBee electronics' highly efficient solar power harvester. At the same time, the device was harvesting energy autonomously from the solar power source when controlling the power point tracking. A power management module with two buffers was used to prolong the system's life. Such systems also played an essential role in the production of a thermal energy harvester. Nevertheless, all the criteria for developing high-efficiency and long-term TEGs for WSNs were not taken into account in those current studies.

2.2 Energy Storage

WSN systems have an essential purpose for long-term service. This is achieved by reducing the system's energy usage, increasing the battery's energy capacity and refilling battery power over time. The reduction in energy can be accomplished by better hardware design and smarter power management, which ensures that unused components are disconnected or slowed down in idle times [70]. Unfortunately, both the energy capacity and the number of recharge cycles of the batteries, even if the device is using

very low energy already, are restricted. As a result, it is essential to not only harvest energy efficiently from the atmosphere but also store the energy in a medium that has far more durability than batteries [71]. The energy storage can be rechargeable batteries or ultra-capacitors. For the selection of rechargeable batteries, the capacity and efficiency are important factors. Further efficiency decreases with charging-discharging cycles, and the temporal energy storage behaviour should be considered. For capacitors, the prime concern is leakage current. The researcher in [72] detailed the properties of conventional batteries used Piezoelectric Energy Harvesting (PEH) applications as a storage element.

2.3 Wireless Network Standards

Wireless networks that are currently available to be used for aircraft health monitoring and automation purposes are numerous. The type of wireless network that one can use is dependent on the intended application because each application has its own set of unique requirements. Given the numerous wireless protocols, selecting a suitable one to use can be a challenge as the different network characteristics get converged and have many commonalities. It is, therefore, necessary to provide an overview of the available wireless network standards that are available in the Industrial, Scientific, and Medical (ISM) band of 2.4 GHz. This frequency band has been reserved for international use for industrial, scientific and medical purposes other than telecommunications. It is the band of choice for wireless standards such as ZigBee, WiFi, Bluetooth and IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN).

2.3.1 ZigBee

ZigBee [57] is a wireless standard that has been specified by IEEE 802.15.4 and maintained by ZigBee Alliance. It is mainly applicable or intended for remote sensing and monitoring applications. It was intended for cost effective, standards-based wireless networking solutions. A good characteristic of the network is that it has a self-healing mesh network and it is self-

reconfigurable if one of the nodes or routes fails. ZigBee network can handle up to 65000 nodes and can provide data rates of up to 250 kbps and has a maximum range of 100 metres lee2007comparative.

Following are some features of ZigBee [73],

- 20 kbps, 40 kbps and 250 kbps data rates.
- Two modes of addressing of Institute of Electrical and Electronics Engineers (IEEE), 16-bit short and 64-bit
- Critical latency supporters, like joysticks
- Carrier Sense Multiple Access/Collision Avoidance channel access.
- Automatic network set up by the controller
- Fully handshake protocol to ensure reliable transfer
- Management of power to ensure low energy consumption

There are 16 channels in the case of the 2.4GHz ISM band, and there are 10 channels in the case of the 915MHz. There is one channel in the case of the 868MHz band.

ZigBee provides the following advantages to Actuator Networks and Wireless Sensors [74]:

- high reliability, the packet loss rate ranges from 0% to 1% for 20m to 90m distance,
- low energy consumption because of using the CC24* transceivers, the transmitting and receiving power ranges from 3mW to 30mW for 10m to 90m distance.
- high-security level which uses 128 bits Advanced Encryption Standard (AES) and use of an open standard

ZigBee is designed and distributed as an open global standard. This provides the possibility for any vendor to implement devices according to the ZigBee specification for a very low price. Furthermore, interoperability is available as long as all vendors comply with the ZigBee specification. It is a big benefit for the customers to have that wide range of ZigBee products available.

2.3.2 Bluetooth

Bluetooth is a wireless technology which is technically known as IEEE 802.15.1 developed by Erickson [75, 76]. It was intended and indeed used for Personal Area Network (PAN) and exchange of data over very short distances [77, 78]. Although Bluetooth nodes can be powered down, they can take up to 3 seconds for them to power back on which can be a disadvantage for some applications such as real-time multimedia communication [79].

Despite having high data rate (up to 3Mbps) compared to ZigBee, Bluetooth has less coverage (up to 10m), low level of security (uses E0 Stream Cipher) and higher power consumption (0-10dBm) [80].

It becomes evident that ZigBee is desirable as a wireless communication protocol for use in WSN systems. It has the most network options from which to choose, and it offers three separate topologies, it is highly scalable and provides different types of modulation schemes which makes it highly immune to noise and interference signals. Furthermore, its low power consumption and moderate speeds make it the wireless protocol of choice for this project. It can be used to build an extensive mesh network of sensor nodes that can be used for system monitoring, and its range also makes it ideal.

2.4 WSN Architecture

The sensor node is composed of a processing unit (Microcontroller unit), a sensing unit, a communications unit and an energy source. This is shown in

Figure 2.3. The first block is a power block which consists of a power battery and DC-DC to power the sensor node. The battery cannot be regularly replaced and therefore, must thus be correctly and effectively used. The second block, which is the communication block, also known as a transceiver, it provides a communication channel in the form of infrared, radio and optical channel. The processing block is the third block. This block has Random Access Memory (RAM), timer, microcontroller and operating system. It is responsible for processing, storage and running events. The sensing unit makes the fourth block. This is made up of several sensors that generate the electric signals by sensing the physical environment and Analog-to-digital Converter (ADC). The fifth block is the software block Software [81].

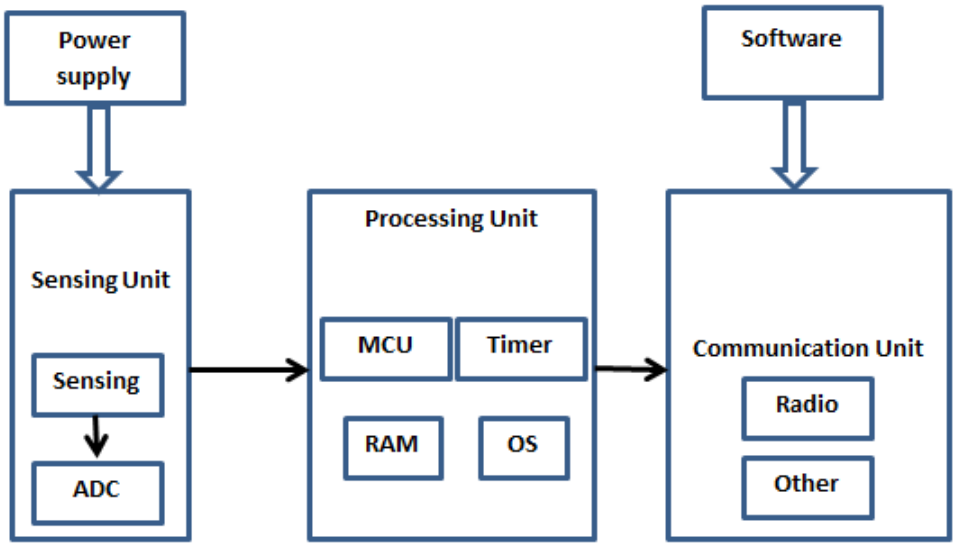

Figure 2.3. WSN architecture

The Processing Unit is responsible for the collection, processing and storage of data from different sources. The central sensor node processor unit determines a node's energy consumption and calculation capabilities. The vast number of microcontrollers, microprocessors, and Field-Programmable Gate Array (FPGA) are required to provide flexibility for the Central Processing Unit (CPU) implementation [82].

The processor used in the processing is the microcontroller. It is composed not only of memory and processor but also of interfaces and non-volatile memory. It helps to reduce cabling, extra hardware, circuit board space and energy requirements. The microcontroller should have three states, sleep, idle and active to conserve power consumption.

A particular clock type is a timer. Since it is asynchronous, it needs a timer to sequence the sequence. Timers include various kinds, such as mechanical, electronic and digital. It also has low energy consumption and low volume. Some WSN node operating systems are TinyOS and Contiki [83]. TinyOS is the best known operating system in the Event-driven sensor network, which calls the correct event manager for execution.

RAM is an internal storage memory for microcontroller information. For programming code storage, flash memory can be used. Memory size can, however, affect energy consumption and increase costs. Thus it is necessary to select the right memory size for a particular application.

The power unit provides energy to the sensor node for low cost and time monitoring of the environment. It takes energy from the electricity generator and goes to a different node component. Sensor node's life depends on the battery. Therefore, the battery is the key component that needs to be properly distributed. For the following reasons, power units are required to ensure long life, voltage stability, load efficiency, low-power charging and low self-disposal. The transmitter and the receiver are on the same circuit board and function as a communication unit. It receives a processing unit command and passes the command on to the other network node. Communication is conducted via wired and wireless communication channels.

Four separate services are provided by the software, and these are: sensor manager, which offers access to sensors and provides transmission of the data to and from the sensor storage is tasked to provide data streams with the persistent storage, iii) query manager which is tasked to process queries and the management of active queries, iv) access control which is provided by the integrity service.

2.5 Routing Protocols

The main objective of the WSNs routing protocol is to decide the best route between source-sink nodes and nodes. At the same time, there should also be other properties in the routing protocol, such as energy conser-

vation, low latency and fault tolerance. And the core concept for protocol routing is power saving. In WSNs, there are some typical routing protocols which are categorized as cluster-based (such as LEACH and Threshold-sensitive Energy-Efficient Sensor Network (TEEN) protocols), data-based (such as Directed Diffusion (DD), Rumor and SPIN protocol), geographic-based (such as Global Positioning System Receiver (GPSR) protocol) and energy-based (such as Energy Aware Routing (EAR) and Distributed Energy Harvesting Aware Routing (DEHAR) protocols). These protocols are described in the following paragraphs.

The network is usually divided into clusters as regards the cluster-based routing protocol. Each cluster has one head cluster and several members of the cluster. Cluster heads organise all nodes within the same cluster, data fusion, and data transmission. These routing protocols include the LEACH protocol [84] and TEEN protocol [85].

Each node is treated equally in the data-based routing protocols. It is a simple and robust protocol, however, It's not very scalable. Furthermore, the data-based routing protocol needs to maintain tables that take a lot of storage space and increase the network's communication output. Typical data-based routing protocols are DD [86], Rumor [87] and Sensor Protocols for Information via Negotiation (SPIN) [88].

Furthermore, only adjacent nodes can rely on geographic protocols such as GPSR protocol [89], so this is almost a stateless. Wireless network nodes often use the shorter distance from Euclidean to create or maintain a routing table, save it and have a short transfer wait.

One of the first protocols suggested for routing wireless sensor networks is the Energy Routing Protocol. The entire network needs global information for routing. However, the node can only access topological information of the local network due to the energy constraints in wireless sensor networks. It is, therefore, just an ideal case. Based on the above, the EAR has been proposed by Shah et al., [90].

A new adaptive and distributed routing algorithm to find an energy-efficient route in a wireless sensor network with energy harvesting is the DEHAR

[91]. The algorithm identifies an energy-efficient path from each node to a single sink node and takes into consideration the current network energy status. The advantages and disadvantages of each type of routing protocol are unavoidable. The selection of the routing protocol should be network-based.

2.5.1 LEACH Routing Protocol

LEACH, which Heinzelman [92] introduced, is a hierarchy of low energy adaptive clusters of the WSN. The LEACH operation can be split into rounds. The clusters will start every round with a setup step followed by a steady-state step. Where multiple data frames are transmitted from the nodes to the head of the clusters and on to the base station. The main goals are to prolong the life of the network, minimise energy use and use data compression to reduce the number of messages during transmission.

To organise the network in several clusters, LEACH adopts a hierarchical approach. Each cluster has a designated head of the cluster. The Head of Clusters is responsible for conducting a variety of tasks. The first task is to regularly collect data from the members of the cluster. The cluster head aggregates it to eliminate redundancy between correlated values after the data are collected. A cluster head's second major job is to direct the aggregated data to the base station. The accumulated data can be transmitted in one hop. The third main task of the head of the cluster is the creation of a Time Division Multiple Access (TDMA) schedule, which will assign each node of the cluster a time frame for the transmission.

The head of the cluster publishes the schedule by broadcasting to its cluster members. LEACH nodes use a Code Division Multiple Access (CDMA) based communication scheme to reduce the likelihood of collisions between sensors both within and outside the cluster. LEACH's basic operations are organised in two stages, (1) the setup stage: the selection of the cluster heads and configuration of the clusters. (2) steady-state phase: a collection of data, aggregation, transmission to the base station (Figure 2.4).

The setup duration is assumed to be comparatively shorter than the steady-state stage to minimise the overhead protocol. At first, each node decides to become a cluster head for the current round or not. It is based on the priorities determined in Bahl et al., [93] and the number of times the node is the head of a cluster.

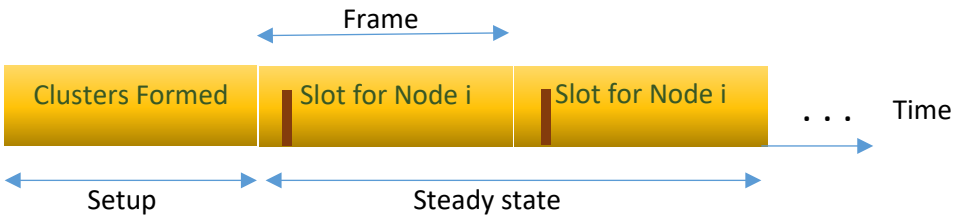

Figure 2.4. LEACH phases

You determine this by selecting the n node, which is the random number from 0 to 1. The node is selected to be a cluster head for the round if it is less than the $T(n)$ threshold. The threshold level is set by (1) [93],

$$T(n) = \frac{p}{1 - p(rmod(\frac{1}{p}))}, if\ n \in G, otherwise = 0 \tag{1}$$

Where G is the set of nodes that in the last rounds were not cluster heads, the p is the desired cluster head percentage, and r is the current round. The nodes that are cluster-heads in the current round advertise message to the rest of the nodes with the same transmission energy. The non-cluster-head recipient must be present during this advertisement process.

Every node then determines to which cluster it belongs based on the signal strength of the signal received. In the case of the ties, random cluster-head is chosen.

LEACH is a cluster-based hierarchic routing protocol [94] for WSN applications. The WSN nodes are divided into clusters. Then the nodes become a hierarchical structure after the clustering. A Cluster Head (CH) node for each cluster is selected that is responsible for scheduling TDMA [95] and sends aggregate data to the cluster Base Station (BS). LEACH is expected to provide sufficient transmitting power for each WSN node. The power enables the CH node to reach the BS or CH node and thus transmission of

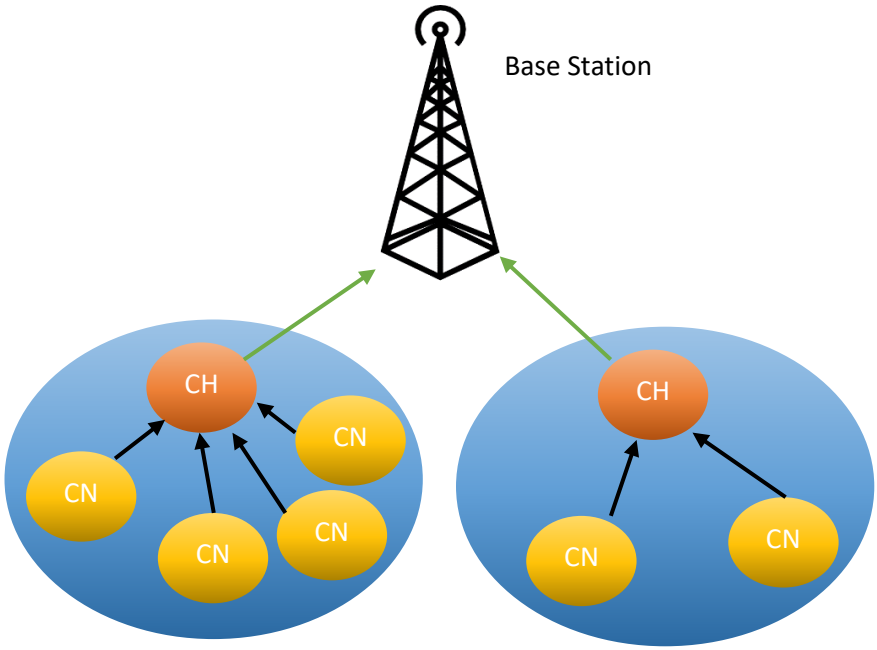

Figure 2.5. Clustering model

data directly into the BS or into its neighbouring CH node. The data can be transmitted directly into the BS node. After deploying and transmitting data to the CH node, as shown in Figure 2.5, the nodes in the LEACH protocol pick their own CH node. With this approach, every node can retain its scarce resources. The additional responsibility as a result of being a CH node will further drain the energy of the nodes acting as CH nodes. By using random rotation techniques in choosing the CH node in the cluster nodes, LEACH protocol alleviates this problem. To prevent cluster collisions, LEACH uses TDMA or CDMA techniques.

The LEACH routing protocol process consists of several rounds. Each of these LEACH protocol rounds consists of Set up and Steady-State phases,

1. Set up phase: During the setup, the CH is randomly selected among the nodes, and various clusters are dynamically formed [96]. Each node initially generates a random number between 0 and 1. This node is chosen as a "CH" during the current round if it is less than a threshold, T(n). This decision also includes CH as the node's history [97].

2. Steady-state phase: The CH node starts assigning its TDMA schedule algorithm to its cluster nodes once the clusters of nodes have been

established. The sensed data is then transmitted to each member node of the formed cluster to its CH node. The CH node will send aggregated information [98] along with its data to the BS cluster once all data have been obtained from its member nodes. The length of the steady-state period is considered to be longer than the setup process. The WSN enters another setup phase after a certain period, ready for another round. After each round has been completed, new clusters will be re-elected for the CH nodes. The lifetime of the WSN can, therefore, be approximated according to the number of rounds completed.

2.6 Power Management System

One of WSN's most significant well-known problems is energy [99]. When a sensor node energy becomes depleted, its functions in the network will no longer take place unless the energy source is replaced or the mechanism for harvesting the energy gap is filled. Battery power is the most significant current source of energy used by sensor nodes, but batteries pose several challenges. First, the current leaks, even if not in service, which consume the battery. Second, extreme weather conditions may break down the batteries, resulting in chemical leaks that may cause various environmental problems [100]. Finally, the energy density of the battery is limited, and this may hinder the operation of the sensor node over a long period [101].

To power a sensor node, its power requirement must be less than or equal to the power harvested from ambient energy. The ideal situation would be that the power requirement of the sensor node to be less than the power generated by the energy harvester module. Anything less and the WSN nodes would not operate properly or not operate at all. Another requirement that should be considered is the efficiency of the energy harvester. Specifically, the ambient energy must ideally be converted to electrical energy with high efficiency so that as little energy as possible is wasted. The choice of components used in the design should be judiciously selected to ensure as much power conservation as possible. The different blocks of the harvester should be designed with energy conservation in mind [101, 102]. Hardware, circuit design and components used, Printed Circuit Board (PCB) design

and software development processes should include processes that ensure optimal use of power.

Power management is a critical constrain of any productive and successful Wireless Sensor Network (WSN) operations because power management determines how efficiently the electrical power is used in WSN. It becomes more critical in aircraft for SHM such as aircraft wings due to their exposure to the unpredictable and harsh environment.

WSN is important for SHM on aircraft, as it continuously consists of tracking key parameters for the expectation of ageing results. The primary aim is to minimise costs and improve mechanical health through predictive maintenance. WSN nodes require adequate energy to relay key-controlled parameters for effective SHM. The harvesting of energy from natural sources, such as thermal and vibrations, is used to power WSN nodes to ensure that adequate energy is sufficient at critical moments of transmission.

Various ambient sources can be used by the WSN nodes to harvest the energy for consumption. TEG and Piezoelectric Energy Generator (PEG) are well known to be dependable sources in the aircraft where energy can be harvested from. Air turbulence and engine vibrations in the aircraft causes vibrations and therefore, placing the PEG devices in the aeroplane wings will effectively harvest the energy. TEG devices placed around the aeroplane engine can also be useful in the energy harvest energy process.

Researchers in [103],[104] and [105] focused on the proposal of piezoelectric generators harvesting techniques with high average strain and low resonant frequency. These PEG techniques were shown to be appropriate to provide energy to the aircraft WSN nodes placed across the wings. The quantity of the harvested energy is dependent on the frequency and amplitude vibrations of the piezoelectric components.

Thermoelectric energy harvesting generators for aircraft were devised by researchers [106]. For TEG to be effective, a thermal gradient is required to harvest the thermal energy [107]. The thermal harvested energy is dependent on the thermal gradient.

A number of researchers have focused on proposing energy harvesting techniques for PEG such as as [103], [104] and [105] with low resonant frequency and high average strain. These techniques are suitable to power low energy WSN nodes in aircraft wings. The vibration amplitude and frequency determine the amount of the energy that can be harvested by the PEG devices.

Power management system with static decisions can lead to energy loss. For instance, because of different operating properties, the WSN nodes drain their energy unequally. For example, energy drainage in the nodes near the sink of data is much faster as other nodes data are transmitted. To match the node's energy consumption and traffic conditions, the battery charging of the node is required. Therefore, intelligent power management decisions are required to save energy while maintaining efficient and successful WSN operations. At the hardware level component selection should ensure that the energy is efficiently harvested, AC/DC and DC/DC converters should be of high efficiency to reduce energy wastage, ensure that storage devices are small and efficient and all components should be low-powered where possible.

At the circuit design level, care must be taken to unnecessarily use power . For instance, one must avoid pull-up resistors and instead use pull-down resistors, capacitors should be high quality and surface mounted to avoid power leakages. PCB design must also focus on power-saving techniques such as ensuring the shortest and lowest inductance path between components, minimising peak-power during start-up and minimising overall inductance. Software implementation should also be in high-frequency oscillators during idle or sleep mode, acceptable use of different operation mode and appropriately configuring the outputs to minimise power consumption.

Several researchers proposed Markov chain models for power management in energy harvesting. Authors in [108] proposed an energy harvesting in WSN with linear topology where they derived the packet loss probability due to channel errors and lack of energy in WSN. The energy harvesting process was modelled as a Markov chain where states represented average harvesting power levels. Although the paper claims to gain near the optimal performance of the proposed model, packet loss probability based on chan-

nel errors and lack of energy did not reflect the realistic scenario. Packet loss due to queuing overflow should also be considered.

Authors in [109] found that Generalized Markov (GM) model was effective in representing PEG harvesting because it is bursty. Authors in [110] proposed the Markov based model that can predict future power consumption and residual availability of energy in WSN. The proposed model captured the energy states of WSN nodes and predicted the probability of a WSN node to fail to detect an event due to lack of energy. The disadvantage of this model is on the scalability because the states are based on the total number of WSN nodes.

The scheduling technique between the grid and the harvested energy based on the two Markov chain models was proposed by researchers in [111]. However, because the grid source does not exist in the aircraft, the model proposed was not appropriate for energy harvesting in the aircraft. In this context, this thesis proposes energy harvesting based on the integration between thermal and vibration because these sources are widely available in the aircraft. The scheduling technique deployed in this thesis was based on the Markov chain model.

As it has been explained in the literature, energy harvesting prediction models are prone to errors, either over-use or under-use of the harvested energy occurs. Therefore, model-free schemes such as an expert system are preferred for power harvesting management strategies. In this context, this thesis proposes a novel expert system based power management for harvesting from vibration energy in the aircraft by using wireless sensor networks.

2.7 Energy Optimization

Several techniques for computational intelligence such as genetic algorithms and neural network, statistical, probabilistic, machine learning and fuzzy control methods have been used widely in WSN for several applications such as communication and error corrections [112].

2.7.1 Fuzzy Logic

Generally, fuzzy logic is deployed to solve the system uncertainties by using fuzzy logic or control, power management system with energy harvesting can be optimized without the need for complete information about it.

Fuzzy logic control system attempts to mimic human reasoning and decision-making methodically and scientifically. It provides a natural method to the implementation of control systems, decision-making and proof frameworks in various industries. Fuzzy logic allows researchers to exploit and make functional the empirical and heuristic knowledge characterised by IF-THEN rules. Professor Lofti Zadeh invented the Fuzzy Logic Systems in 1965, and since then it has been used for various scientific, engineering and business applications, such as in smart control systems, error detection, decision-making and expert systems. Fuzzy control systems have also been used in several disciplines such as in engineering and natural sciences [113, 114, 115, 116, 117]. For example, medical analysts manage the consciousness, unconsciousness, pain and its relief, muscle activity and relaxation over a specified period. The anaesthetized patient is placed in the "feedback circuit" for the remainder of the procedure in the operating theatre. The system that has been designed with fuzzy will be able to decide by using fuzzy logic. A system designed with fuzzy control will be composed of algorithms that have inference. The algorithm with inference will deploy IF-THEN rules with AND/OR connections.

Fuzzification is the first step of the fuzzy inference. The crisp sets (e.g., Yes or No) are transformed into fuzzy sets (e.g., values between 0 and 1) by the fuzzification. This process increases the system's uncertainty and uses the membership functions to link crisp sets with fuzzy sets.

Defuzzification is the mechanism by which a single number is derived from the output of the combined fuzzy set. The fuzzy inference results are transmitted as crisp out. In other words, defuzzification is achieved by a decision-making algorithm that selects the best value based on a fuzzy set. Defuzzification is the way that a fuzzy set with a small number is portrayed with a crisp number. Internal data representations in a fuzzy system are usually portrayed as fuzzy sets. However, the output often needs to be a crisp

number that can be deployed to perform a function such as controlling the WSN power management system to charge the battery. Figure 2.6 shows the fuzzy control system and its components.

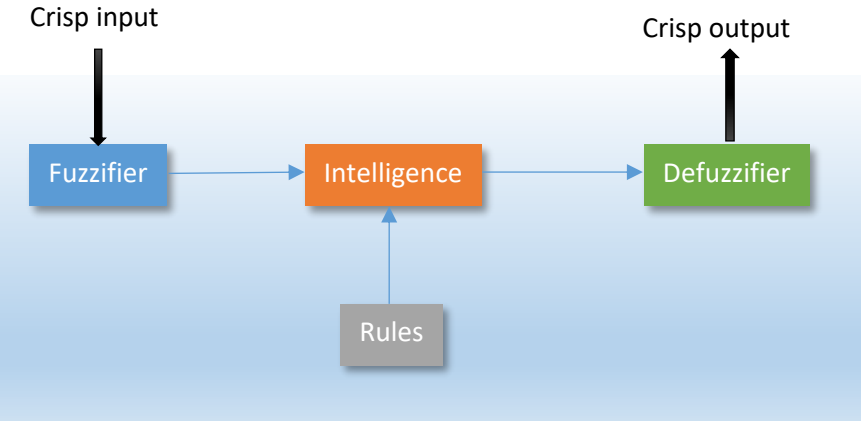

Figure 2.6. Fuzzy control system

2.7.2 Machine Learning

Machine learning is an Artificial Intelligence (AI) technology that enables systems to learn and develop automatically without being specifically programmed. Machine learning focuses on the development of computer programs that can access and use data to learn for themselves.

The learning process begins with data observations, such as examples, direct experience or feedback, to find trends in the data and, based on the examples we present, to make better decisions in future. The primary goal is to ensure that computers automatically learn and adjust behaviour without human intervention or help.

Algorithms for machine learning are also categorised as supervised or unsupervised [118].

- Machine learning supervised algorithms can use labelled examples to predict future events to apply what was learned in the past to new data. The learning algorithm produces an inferred function, which starts with the study of a known training dataset, to predict the output values. After sufficient training, the system can provide targets for any new input.

The learning algorithm may also compare its output to the appropriate output and recognise errors to change the model as required.

- Unsupervised machine learning algorithms, by contrast, are used if data used for training is neither labelled nor labelled. Unsupervised studies how systems can derive a function from unlabelled data from describing the hidden structure. The system does not find the right output, but it explores the data and can draw data sets to describe hidden, unlabelled data structures.

- Semi-supervised algorithms are between supervised and unsupervised learning. Since they both use labelled and unlabelled training data, typically a small amount of labelled data and a large amount of unlabelled data. The systems using this approach will greatly increase the precision of learning. Semi-supervised learning is usually chosen, when skilled and relevant sources are required to train/learn from acquired labelled information. If not, it does not usually require additional resources to obtain unlabelled knowledge.

- Reinforcement machine learning algorithms is a method of learning that communicates with its environment by generating behaviour and finding errors or rewards. The most relevant characteristics of enhanced learning are trial and error search and delayed reward. To optimise its efficiency, machines and software agents may automatically evaluate the perfect behaviour within a particular context. For agents to learn what is best; this is known as the reinforcement signal, simple reward feedback is required.

Many of the WSN applications use fast, secure and effective data communication to a large extent. However, because of its loosely connected nature, WSN communication connections are inherently unreliable. Consequently, communication protocols adapt situationally so that healthy and energy-efficient routes can be identified. A guided teaching approach for routing optimization was developed by [119] and [120] authors. Machine learning techniques are used to increase situational awareness to optimise communications in the algorithm. It primarily used machine learning to detect correlations between input and network-level, buffer occupancy, packet

length, residual node energy and output (e.g., link quality, optimal way) features automatically.

Machine learning allows for extensive data analyses. Although the results are usually faster and more accurate, it is therefore not energy-efficient since additional time and resources can be required to train it properly. Consequently, in this thesis, the Fuzzy control method is preferred because it requires fewer resources and, therefore, energy efficiency.

2.8 Simulation Tools

Analytical, experimental or by a combination of these methods may be used to study the behaviour of a system. However, analytical methods cannot detail the impact of power consumption in full. Experimental research can provide more precise information, but this is done at a higher cost. Simulation models, therefore, provide the best alternative to the low-cost and short-term understanding of the behaviour of a system.

There is a wide range of simulation platforms available, but only a few simulators could be used for certain operations. This thesis has deployed LTspice and NS3 because they are widely used by the research community for both communication and circuit simulation purposes.

2.8.1 Network simulators

NS2: Network Simulator 2 is a joint initiative by people from the University of California in Berkeley, the University of Southern California, the Lawrence Berkeley National Laboratory and Xerox Palo Alto Research Center and was developed under the VINT (Virtual InterNetwork Testbed) Project in 1995 [121].

The Defense Advanced Projects Agency and the National Research Foundation are the key supporters of this initiative. It is a discrete event simulator that offers significant support for the simulation by wired and wireless net-

works of TCP, routing and multicast protocols. NS2 is a network simulator based on the object-oriented paradigm, which was initially developed at the University of California-Berkely and is powered by discrete events. The programmes it uses are C++ and OTcl (the language of Tcl scripts with MIT object-oriented extensions). There is a justification for using these two programming languages. The main explanation is that these two languages have their internal characteristics. C++ is designed effective, but visual and graphical visibility is not very fast. It's not easy, without a very visual and concise language, to adjust and assemble different components and to alter different Parameters.

NS3: The NS3 Simulator is a discrete network event simulator developed primarily for the research in computer networking protocols, and the educational use purposed [122]. The NS3 is an open-source project that was launched in 2006. The GNU GPLv2 licenced NS3 as free software. It depends on the ongoing group efforts in creating new models, testing or improving them and sharing information. In NS3 and NS2, the main difference includes, the NS3 core part is coded in C++, and the scripting interface is written in Python. NS3 supports lightweight virtual machines such as Linux containers.

2.8.2 Circuit simulators

Simulation Program with Integrated Circuit Emphasis (SPICE) was first developed at EE Berkeley. Mathematical models are used to describe circuit elements. The LTspice is widely used by researchers in electronics to simulate electronics circuits [100].

2.9 Summary

This chapter has reviewed the state of the art research on energy harvesting technologies. The main covered topics on energy harvesting were piezoelectric and thermal harvesters. These two energy harvesting sources have been used in this thesis because they are widely available in the aircraft.

This chapter has also reviewed current WSN power management systems together with intelligent systems such as fuzzy logic and machine learning. A wide range of simulation tools was also reviewed, including NS3 and LTspice for communication and circuit design simulation tools.

This thesis has recognised that there is a gap in the power management system for efficient energy harvesting from vibration and thermal sources in AHM because the current systems are based on prediction models which have shown to be prone to errors, either over-use or under-use of the harvested energy occurs. The expert based system has been proposed in this thesis to bridge this gap and hence, efficient energy harvesting model has been proposed.

There is a gap in the deployment of simulation tools because the current popular tool for electronics (LTspice) and communication (NS3) are used in isolation. This study has integrated these two popular tools in simulations and proved that through virtualization a communication channel (TCP or UDP) between LTspice and NS3 could be created, and parameters such as harvested power can be output from LTspice and be communicated to the NS3 simulation in real time.

There is a gap in the deployment of energy-efficient routing protocols. The existing power management systems have been deploying routing protocols that contribute to high energy consumption. This study proposed to deploy the LEACH routing protocol in AHM systems that proved to efficiently conserve more energy and hence prolong the lifetime of the wireless sensor nodes.

3 PEG and TEG INTEGRATION

This chapter proposes the integration of PEG and TEG into the aircraft environment to guarantee a sufficient level of energy in WSN nodes. The uncertainty of efficiency and adequate resources for data transmission makes this solution an essential step. This uncertainty will lead to intolerable delay and even loss of data.

Efficient scheduling to reduce data losses and minimise power consumption by the communication channel is necessary for the integration of PEG and TEG. The thesis, therefore, suggests a Markov Chain model for a scheduling system which would provide the real-time decision based on the current state of energy usage, energy storage and data queue. The Simulation based on NS3 will implement, evaluate and validate the scheme.

3.1 System Model

The energy storage capacity, packets queue length and WSN power consumption are used by the scheduling scheme and make the real-time decisions on which harvesting source should be selected to charge or recharge the energy storage. The system model is depicted in Figure 3.7. The system model has four main functions. The scheduling scheme is tasked with interpreting energy storage capacity status (such as full or empty), WSN power consumption (such as high or low) and packet queue length (such as full or empty) and decide by switching power harvesting between PEG and TEG ambient energy harvesting sources. The packets queue length is obtained from the WSN receiving node. Communication, processing and sensing units are described in Chapter 2 under WSN architecture.

The packet loss probability, power consumption and energy storage level will be derived by the discrete-time Markov chain model. These parameters will then be used for scheduling purposes.

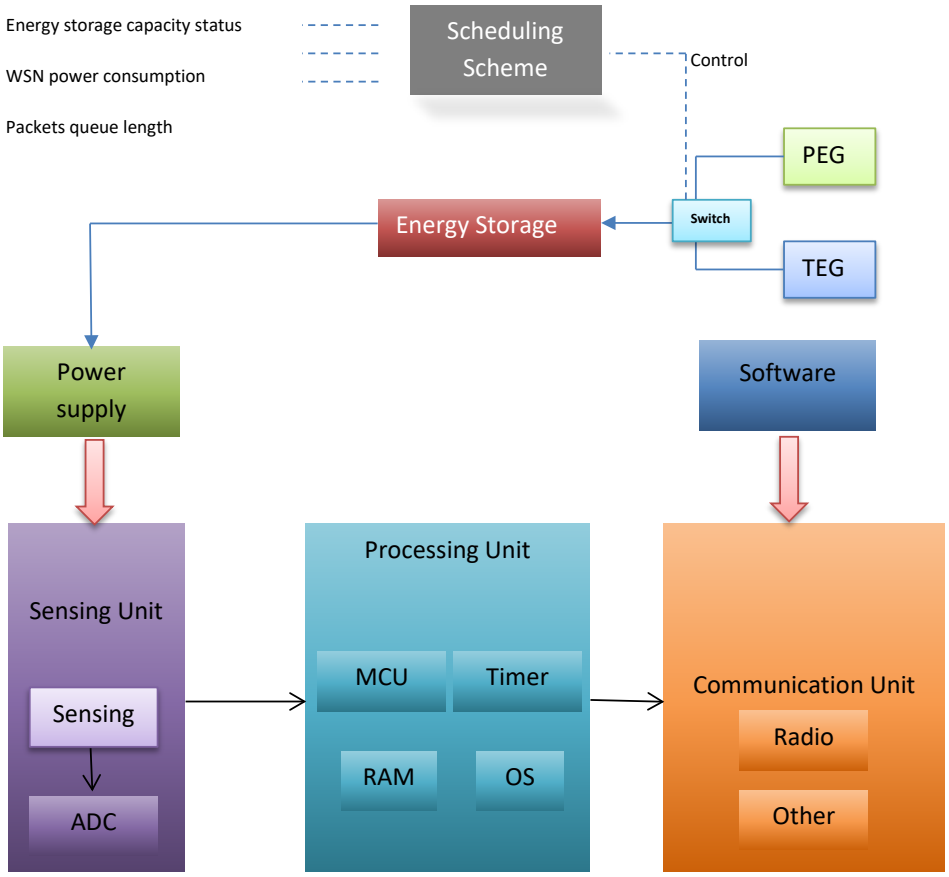

Figure 3.7. System Model

The proposed system model considers the following assumptions,

1. The piezoelectric generator can harvest an infinite quantity of energy
2. The piezoelectric generator energy arrives at a WSN node at random times
3. The thermoelectric generator can harvest an infinite quantity of energy
4. The thermoelectric generator energy arrives at a WSN node at random times
5. The piezoelectric generator or thermoelectric generator can only be used one at a time
6. WSN node energy storage has a limited storage capacity
7. WSN node energy storage is assumed to have enough energy before starting communication

8. Once energy storage capacity has reached the minimum threshold, the transceiver will be put into sleep
9. The transceiver is assumed to be the most energy consuming component in WSN node
10. The simulation is run in a time-slotted manner

The statement (WSN node energy storage is assumed to have enough energy before starting communication) may lead to the unwanted average queueing time and packet losses could therefore occur. Both PEG and TEG could minimise the result based on the current power storage of the WSN node, but this requires an efficient scheduling system.

At each time slot t_i, where i is the slot number, data packets arrive on the receiving WSN node's data queue whose capacity is Q in a manner that is Poisson distributed with the arrival rate of λ_i and probability of p_{di} where d is the data packet. The rate at which these packets are processed follows an exponential distribution with the service rate of μ_i. In this thesis, each packet is considered to be equal in size, and the queue discipline follows the First COME First SERVED (FCFS) model. This data model is implemented as an $M/M/1$ where M stands for Markovian with one processing unit and one server queuing system. This model is well explained in Little's Law [123]. The queue system is further depicted in Figure 3.8,

$$L = \lambda W \tag{2}$$

Where the average number of packets in the WSN node queue is denoted by L, and the average waiting delay in the WSN node queue is represented by W.

$$L = \rho(1 - \rho) \tag{3}$$

Where the utilization of the WSN node processing is denoted by ρ and is calculated as,

$$\rho = \lambda/\mu \tag{4}$$

The average delay time experienced by any packet in the WSN node queueing system and at any time slot t_i is given by,

$$W = \rho\mu(1-\rho) \tag{5}$$

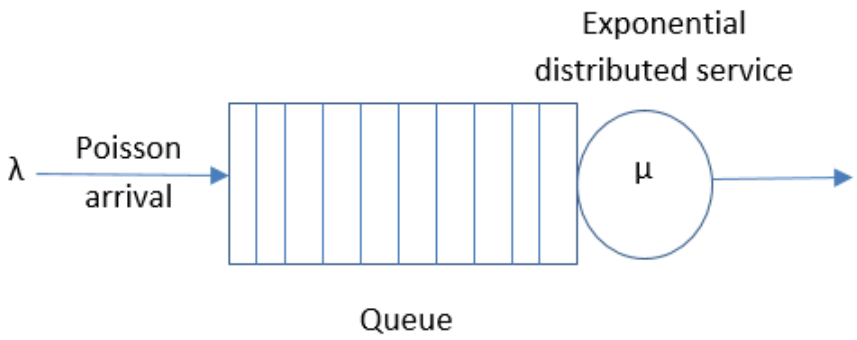

Figure 3.8. M/M/1 queue

3.2 Harvested Energy

The harvested energy by either the piezoelectric or thermoelectric arrives at the energy source in a random fashion. The energy e_i in Joules is assumed to arrive with probability $p_e i$ at the source WSN node has the capacity of communication with n_i number of packets at any time slot t_i.

$$e_i = n_i e_i \tag{6}$$

Where the unit energy is denoted as e_i, which is the quantity of energy required to transmit one packet of data in the WSN node.

If the energy storage capacity of a WSN node is denoted by E, then queueing model $M/M/1$ is also implemented for the energy storage. In this model, the energy is rejected when the storage is full, denoted as $E(e_i)$. The Little's law is also used to compute the average length of the energy queue as,

$$E_{qi} = E - \rho \tag{7}$$

The discrete-time Markov chain is used to model the queues for data packets and energy units. In this model, each state is represented by queue status. For $d \geq 0$ and $e \geq 0$, (d, e) will symbolise the state in the WSN node that the energy queue length is e and the data queue length is d. Furthermore, $\pi(d, e)$ will represent the steady-state probability of state (d, e). The average data queuing delay is also calculated by the Little's law as,

$$D = \frac{\lambda_i}{p_{di}} \sum_{d=1}^{Q} \pi_d \tag{8}$$

The following equation is also used to compute the average energy queueing delay,

$$D = \frac{1}{n_i pe_i} \sum_{i=1}^{E} e\pi_i \tag{9}$$

The proposed model for PEG and TEG system assumes that the TEG and PEG in the aircraft is active at all the time. Both PEG and TEG may be in any of harvesting level states, LOW (L) or HIGH (H).

The High (H) state of the energy harvesting occurred when the value of the PEG or the TEG harvested energy has reached a pre-set threshold. It is at Low (L) state if it is below a pre-set threshold.

Two states (High (H) and Low (L)) of the energy level can also be considered for energy storage (ES). The High (H) state of the energy storage occurred when the value of the energy storage has reached a pre-set threshold. It is at Low (L) state if it is below a pre-set threshold.

Two states (High (H) and Low (L)) of the queue length can also be considered for data queue (D). The High (H) state of the data queue occurred when the value of the queue length has reached a pre-set threshold. It is at Low (L) state if it is below a pre-set threshold.

If the value of the queue length has reached a pre-set threshold, the data queue state is at High state (H). If it is below a pre-set threshold, then It is at Low state. In this thesis, (p, t, s, d) denotes the state of the WSN node where the harvesting state for PEG, TEG, the energy source state and the data queue is $p, t, s, d,$ respectively. The steady-state probability for the state (p, t, s, d) is represented by $\pi(p, t, s, d)$.

The outlined parameters in this section will be used in the simulation, which will involve stochastic scheduling scheme.

3.3 Stochastic Scheduling

The concept of stochastic scheduling is inspired by priority assignment problems in different systems to access common service resources with random characteristics (i.e. arrival time or processing time). In the case of the proposed Markov base model for energy harvesting, the energy storage capacity, packets queue length and WSN power consumption are used by the stochastic scheduling scheme to make decisions on which harvesting source should be picked up to charge or recharge the energy storage. The following 16 cases are observed in simulation for PEG, TEG, ES and D states.

Case 1: If the PEG is in High state, the TEG is in High state, the ES is in High state and D is in High state at any time slot t_i, then data in the data queue will be transmitted

Case 2: If the PEG is in High state, the TEG is in High state, the ES is in High state and D is in Low state at any time slot t_i, then the transceiver will be put to sleep

Case 3: If the PEG is in High state, the TEG is in High state, the ES is in Low state and D is in High state at any time slot t_i then select either the PEG or the TEG with high energy and start energy harvesting. Put the transceiver to sleep

Case 4: If the PEG is in High state, the TEG is in High state, the ES is in Low state and D is in Low state at any time slot t_i then select either the PEG or the TEG with high energy and start energy harvesting. Put the transceiver to sleep

Case 5: If the PEG is in High state, the TEG is in Low state, the ES is in High state and D is in High state at any time slot t_i then transmit the data in the data queue and stop energy harvesting

Case 6: If the PEG is in High state, the TEG is in Low state, the ES is in High state and D is in Low state at any time slot t_i then put the transceiver to sleep and stop energy harvesting

Case 7: If the PEG is in High state, the TEG is in Low state, the ES is in Low state and D is in Low state at any time slot t_i then put the transceiver to sleep and start energy harvesting by selecting PEG

Case 8: If the PEG is in High state, the TEG is in Low state, the ES is in Low state and D is in High state at any time slot t_i then put the transceiver to sleep and start energy harvesting by selecting PEG

Case 9: If the PEG is in Low state, the TEG is in Low state, the ES is in High state and D is in High state at any time slot t_i then transmit the data in the data queue and stop energy harvesting

Case 10: If the PEG is in Low state, the TEG is in Low state, the ES is in High state and D is in Low state at any time slot t_i then put the transceiver to sleep and stop energy harvesting

Case 11: If the PEG is in Low state, the TEG is in Low state, the ES is in Low state and D is in High state at any time slot t_i then select either the PEG or the TEG with high energy and start energy harvesting. Put the transceiver to sleep

Case 12: If the PEG is in Low state, the TEG is in Low state, the ES is in Low state and D is in Low state at any time slot t_i then select either the PEG or the TEG with high energy and start energy harvesting. Put the transceiver to sleep

Case 13: If the PEG is in Low state, the TEG is in High state, the ES is in High state and D is in High state at any time slot t_i then transmit the data in the data queue and stop energy harvesting Case 14: If the PEG is in Low state, the TEG is in High state, the ES is in High state and D is in Low state at any time slot t_i then put the transceiver to sleep and stop energy harvesting

Case 15: If the PEG is in Low state, the TEG is in High state, the ES is in Low state and D is in High state at any time slot t_i then put the transceiver to sleep and start energy harvesting by using TEG

Case 16: If the PEG is in Low state, the TEG is in High state, the ES is in Low state and D is in Low state at any time slot t_i then put the transceiver to sleep and then start energy harvesting by using TEG

One of the above described cases will be observed at any time slot t_i in the simulation.

3.4 Simulation Setup

Analysis, experimental or a simulation model or a combination of these methods may be used to analyse the proposed method. Nevertheless, analytical approaches can not explain the effect of power usage in full. Experimental research can provide more accurate details, but it can be done more expensively. Simulation models, therefore, provide the best alternative to the low-cost and short-term understanding of the behaviour of a system. The proposed system is based on the enhanced energy harvesting framework NS3. Instead of existing sources, this enhanced system involves the deployment of TEG and PEG. The framework has a Device Energy Model, Energy Source, and Energy Harvester (c.g., Figure 3.9). The sample code is appended in Appendix A.

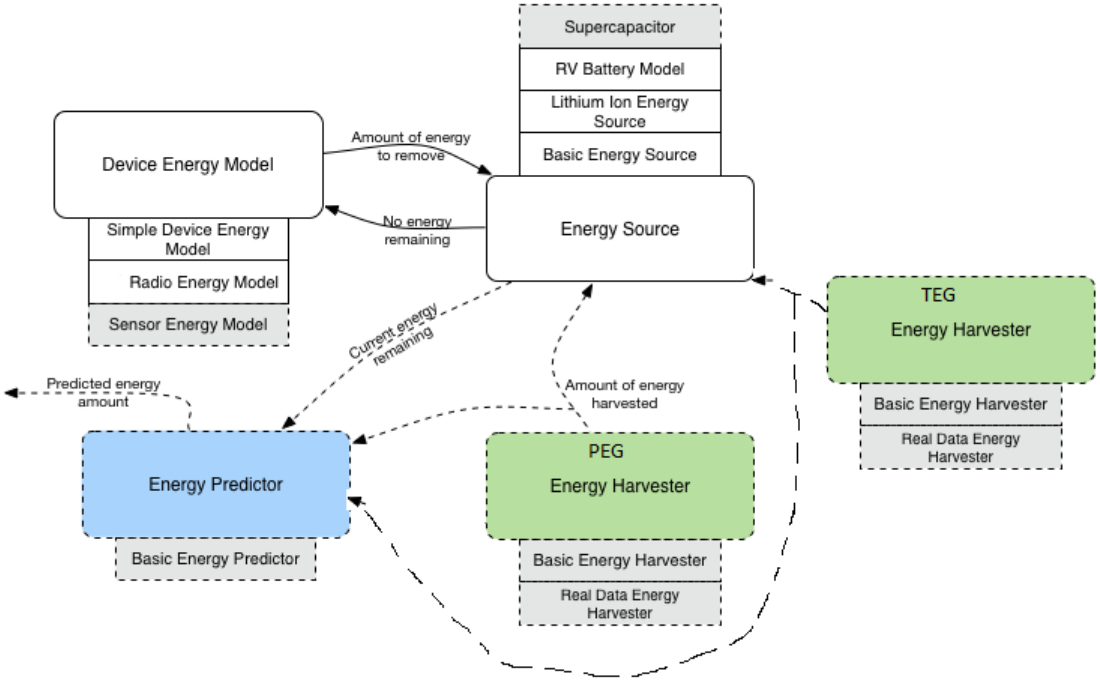

Figure 3.9. Energy Harvesting Framework [2]

The power supply for each WSN node in the proposed system is denoted by the energy source. It is assumed that at least one energy source may be attached to the WSN node, and several energy models can be linked to each energy source. The relation of the WSN node energy source to an energy model of the equipment means that the equipment draws power. The primary functionality of the energy source is to supply the energy to the WSN node. If power is drained entirely from the WSN energy source, all the

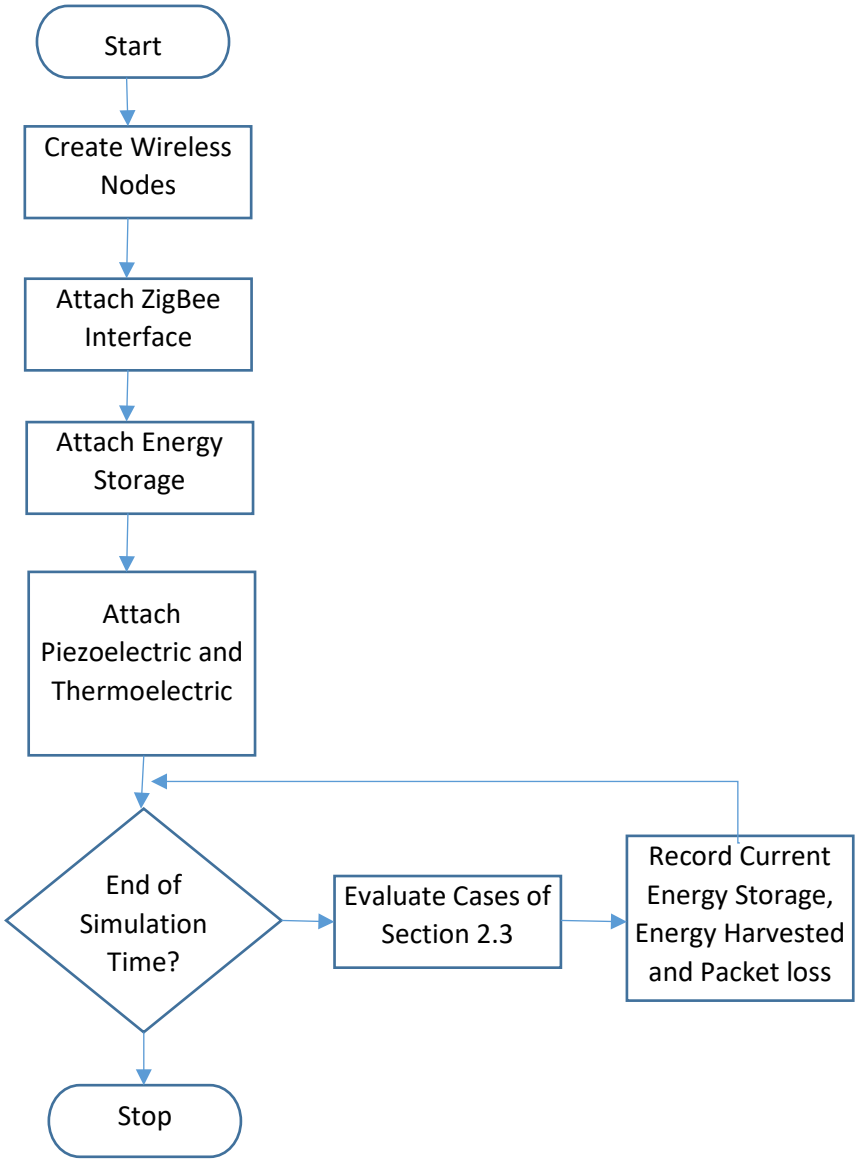

Figure 3.10. Simulation flowchart

devices on the WSN are notified, so that every WSN node can respond to the event. Moreover, for information like remaining energy or power factor (battery level), each WSN node is allowed to access the Energy Source Objects.

The Device Energy Model is the model for a system mounted in a node that consumes power. It is designed as a state-based model, where every device is supposed to have multiple states, and every state has a power consumption value. The corresponding Energy Model of the device must notify the energy source of the new current drawing of the device if the device state

changes. The energy source calculates and updates the remaining energy for the new total current draw-up.

The Energy Radio Model is a radio network device's energy consumption model. It offers an individual state of Idle, Tx, Rx, Sleep and Off for each of the available States. Each of these states is related to the current draw (in Ampere). At each transition, energy consumption is calculated in the previous state, and the remaining energy source is notified.

The flowchart to depict the simulation of the PEG and TEG based on Markov chain is depicted in Figure 3.10.

Algorithm 1 Sumulation algorithm: PEG and TEG integration

```
1: initialize maxEnergyPiezo = 0.01;
2: initialize minEnergyPiezo = 0.003;
3: initialze rEnergy = 10.0;
4: initialize maxEnergyLevel = 0.15;
5: initialize minEnergyLevel = 0.11;
6: initialize iniEnergyLevel = 0.13;
7: initialize minQSize = 200;
8: initialize maxQSize = 300;
9: initialize receiverQueue;
10: initialize receiverQueue;
11: start simulation timer;
12: set Prss = -80i dBm;
13: set PpacketSize = 200 bytes;
14: set number of = 1000000;
15: set interval = 0.02 seconds;
16: set harvestingUpdateInterval = 1 second;
17: create wsn nodes;
18: attach ZigBee protocol to wsn nodes;
19: create ZigBee channel MAC layer;
20: attach energySource to wsn nodes;
21: attach energyHarvesters to wsn nodes;
22: schedule simulation with startTime, GenerateTraffic, source, Ppacket-
    Size, wsnnodes, numPackets, interPacketInterval;
23: function ReceivedPacket()
```

```
24:     print received packets;
25:     return
26: function QueuePacket()
27:     if receiverQueue size <= qSize then
28:         add packet to receiverQueue;
29:     else
30:         drop packet;
31: function QueueState()
32:     if queueSize <= minQSize then
33:         set queueState to 0;
34:     else
35:         set queueState to 1;
36: function printQueueSize()
37:     print receiverQueue size;
38: function DequeuePacket()
39:     if receiverQueue is not empty then
40:         remove packet from receiverQueue;
41: function ReceivePacket()
42:     while there are more packets do
43:         QueuePacket();
44: function GenerateTraffic()
45:     if pktCount > 0 then
46:         Send packet;
47:         schedule (GenerateTraffic, socket);
48:     else
49:         Close Socket;
50: function PutRadiotoIdleandSleep()
51:     put basicRadioModelPtr to sleep;
52: function PutRadiotoActive()
53:     put basicRadioModelPtr to active;
54: function HarvestedState()
55:     if harvestedPower >= minEnergyPiezo && harvestedPower <=
    threSholdPower then
56:         harvestState = 0;
57:     elseharvestState = 1;
58: function CurrentEnergyState()
```

```
59:     if currentEnergy <= minEnergyLevel  then
60:         energyState = 0;
61:         PutRadiotoIdleandSleep();
62:     if currentEnergy > minEnergyLevel then
63:         energyState = 1;
64:         PutRadiotoActive();
65: function HandleEnergyRecharged()
66:     Recharge storage
67: function RadioState()
68:     radioState = 0;
69: function HandleEnergyDepleted()
70:     Handle Depleted Energy
71: function setMaxEnergyLevel()
72:     set MaxEnergyLevel
73: function RemainingEnergy()
74:     print remainingEnergy;
75: function TotalEnergy()
76:     print totalEnergy;
77: function HarvestedPower()
78:     print harvestedPower;
79: function TotalEnergyHarvested()
80:     print TotalEnergyHarvested;
81: function radioTransProb(x,y)
82:     compute transProb[3][3]
83:     return transProb[x][y];
84: function CurrEnergyTransProb(x,y)
85:     compute transProb[3][3]
86:     return transProb[x][y];
87: function HarvestedTransProb(x,y)
88:     compute transProb[3][3]
89:     return transProb[x][y];
```

3.4.1 Simulation Parameters

The simulation parameters are configured to depict the distribution of WSN nodes on the aircraft and the total number of packets sent in an interval of 20ms.

- The maximum number of packets to send is set to 100000 packets
- The arrivals or departures of the packets are set 20ms
- The total WSN nodes number is set to 2
- 20ms is chosen as the time slot
- The simulation is started at time 0 s
- The simulation duration is set to 600 s
- 20 m is set as the minimum distance between nodes
- 0.11J is set as the threshold battery level for recharging the storage capacity
- 0.15 J is set as the threshold for maximum battery level capacity =
- The IEEE protocol for wireless communication is chosen as 802.15.4 with transmission and receiving power as TX = 0.0174J and RX = 0.0197J, respectively
- The piezoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.0022 J - 0.0044 J]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [34, 35].
- The thermoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.003 0.22]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [36]
- 300 number of packets are set as the queue size capacity. If this threshold is reached, then packets will be dropped and considered lost.

3.5 Simulation Results

3.5.1 PEG Results

This section presents results when PEG is the only source of energy. It will be compared to TEG and then to the integrated PEG and TEG in the later sections.

Table 3.3 and Figure 3.11 illustrate the energy Storage transition probabilities during the current energy storage states.

Table 3.3. State transition probabilities of the current energy storage

	Low	High
Low	0.92	0.08
High	0.36	0.64

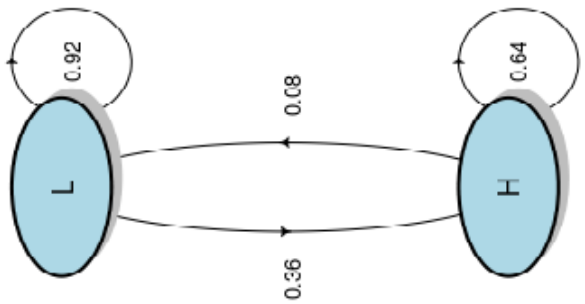

Figure 3.11. State transition probabilities of the current energy storage

The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states. The current energy storage state probabilities results have shown that the High-to-High transition status is at roughly 64%. It can be observed that around 72% of the simulation time, the storage energy status is in High state. It is because, from the simulation parameters, 0.11 J was the threshold for the low current energy. Otherwise, if the current

energy quantity is higher than 0.11 J, it is deemed high. 0.15J was set as the threshold for the maximum energy capacity of the storage. The range of these values were the only possible values for the energy storage model in NS3. The transceiver is put to sleep in order to save energy if the energy storage status is Low.

Figure 3.12 depicts the current energy storage values during the 600 s duration of the simulation. The storage is recharged by using the piezoelectric generator if the minimum energy level is attained due to transmission activities of the transceiver. It can be observed that 0.136566 J and 0.107097 J were the maximum and minimum energy quantities, respectively, reached during the simulation time at each time interval.

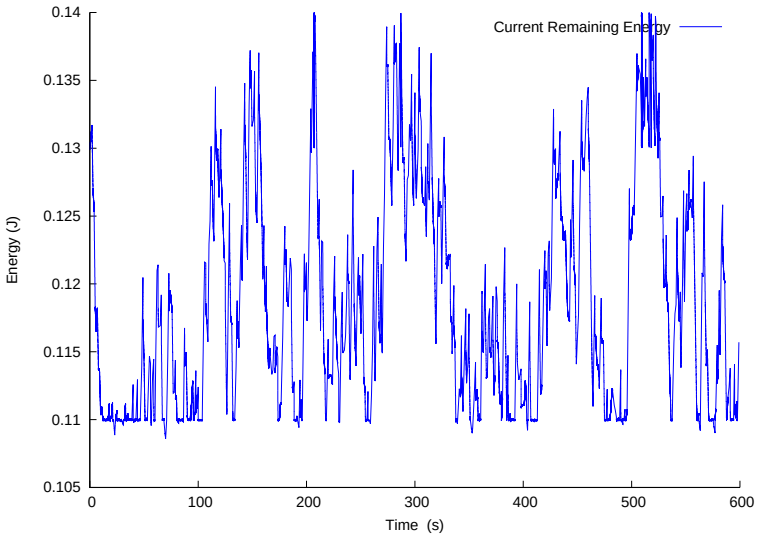

Figure 3.12. Current energy storage values

Table 3.4 and Figure 3.13 show the transition probabilities of the harvested energy states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states. The current harvested energy state probabilities results have shown that roughly 57%, the harvested energy was at low status during the simulation time. The threshold of 0.0033 J was chosen as the low state of the current harvested energy. If the quantity of the harvested energy is higher than 0.0033 J, it is then deemed high. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states.

Table 3.4. State transition probabilities for the harvested energy

	Low	High
Low	0.57	0.43
High	0.27	0.73

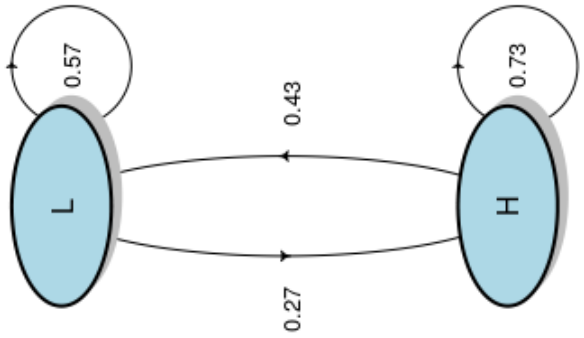

Figure 3.13. State transition probabilities for the harvested energy

Figure 3.14 depicts the current harvested energy values during the 600 s duration of the simulation. The energy storage is recharged if the piezoelectric generator state is high. It can be observed that 0.00438774 J and 0.00014 J were the maximum and minimum harvested energy quantities reached during the simulation time at each time interval. It can be noted that if the threshold harvested energy of less than 0.0022J is reached, then no recharging activities would take place.

The receiving and transmission data queues play a vital role in the WSN nodes. This is because they store data before any transmission and before processing when receiving. Data queues can be congested and result in data loss if the WSN energy storage is at a low state. The transceiver is put to sleep if the WSN energy storage is at the low state. Hence, data loss probabilities and the queue length prediction is vital to prevent data loss in WSN nodes.

Table 3.5 and Figure 3.15 show the transition probabilities of the queue states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation

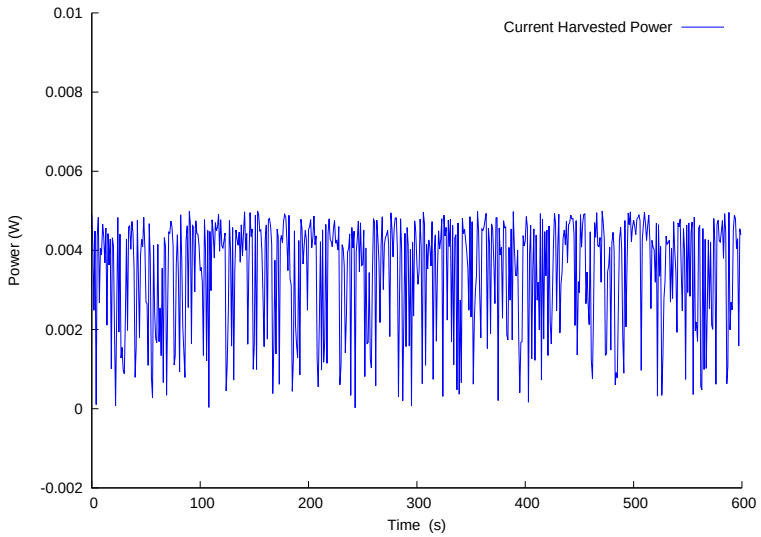

Figure 3.14. Current harvested energy

by recording the ratio of how many times a state has occurred over the total number of recorded states.

The current data queue state probabilities results have shown that the High to High transition status is at roughly 48%. It can be observed that around 99% of the simulation time, the data queue status is in High state. This is because, from the simulation parameters, the number of 300 packets was the threshold for the maximum queue capacity, and 200 number of packets was set for the high queue state.

Table 3.5. State transition probabilities for the queue size

	Low	High
Low	0.49	0.51
High	0.52	0.48

The current queue size during the simulation time is illustrated in Figure 3.16 where the number of 300 packets are pre-set as the maximum queue capacity. If the queue is full, then the packets will be lost. It can be observed that the number of 191 packets was the average queue size. The minimum queue size was empty (the number of data packets were zero).

In the simulation, 7341 was the number of packets that were expected to arrive at the queue for transmission in the duration of the simulation. The

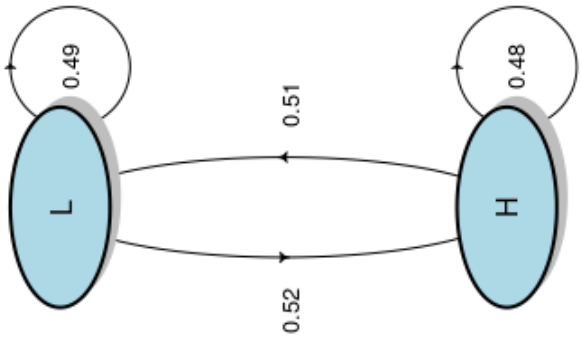

Figure 3.15. State transition probabilities for the queue size

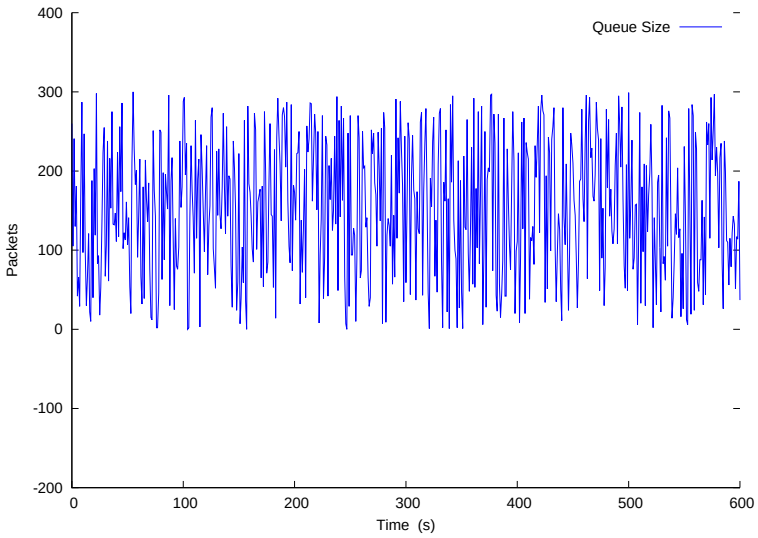

Figure 3.16. Queue Size

successful number of transmitted packets were 1066. However, 5976 number of packets were dropped, and only 299 number of packets remained in the WSN node queue when the simulation completed. This roughly 81% of the packets were lost. Since the data queue was overflowing when the transceiver was put to sleep due to the low energy storage status, high packet loss probability occurred.

3.5.2 TEG Results

Table 3.6 and Figure 3.17 depict the transition probabilities of the current energy storage states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states.

The current state probabilities results of the energy storage have shown that the High- to-High transition status is at roughly 76%. It can be observed that around 85% of the simulation time, the storage energy status is in High state. This is because, from the simulation parameters, 0.11 J was the threshold for the low current energy. Otherwise, if the current energy quantity is higher than 0.11 J, it is deemed high. 0.15J was set as the threshold for the maximum energy capacity of the storage. The range of these values were the only possible values for the energy storage model in NS3. The transceiver is put to sleep in order to save energy if the energy storage status is Low.

Table 3.6. State transition probabilities of the energy storage

	Low	High
Low	0.91	0.09
High	0.24	0.76

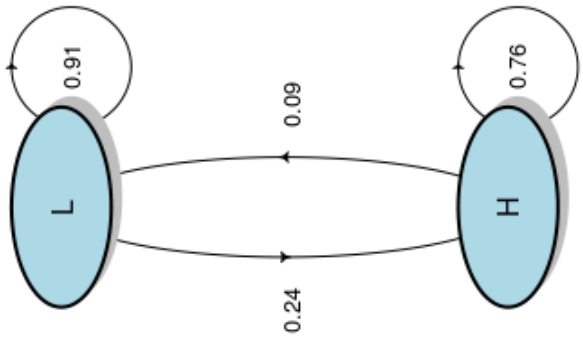

Figure 3.17. State transition probabilities of the energy storage

Figure 3.18 depicts the current energy storage values during the 600 s duration of the simulation. The storage is recharged by using the thermoelectric generator if the minimum energy level is attained due to transmission activities of the transceiver. It can be observed that 0.156588 J and 0.108596 J were the maximum and minimum energy quantities reached during the simulation time at each time interval.

Table 3.7 and Figure 3.19 illustrate the state transition probabilities of the harvested energy values.

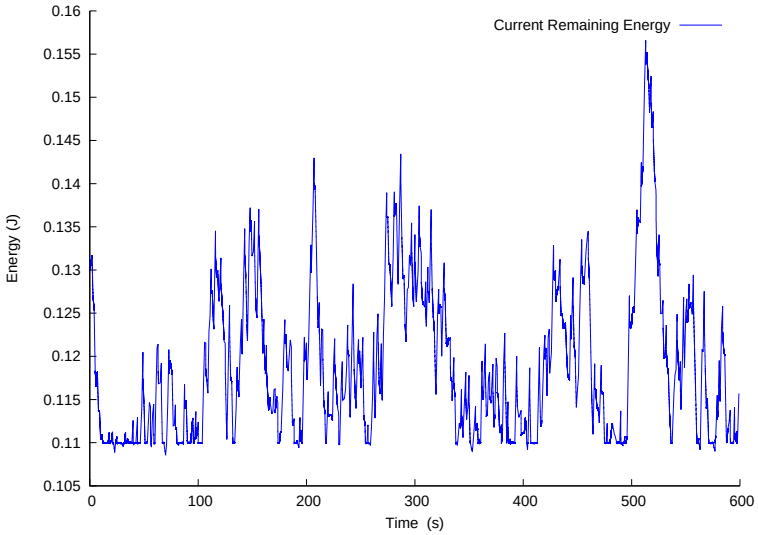

Figure 3.18. Current energy storage values

Table 3.7. State transition probabilities of the harvested energy

	Low	High
Low	0.39	0.61
High	0.42	0.58

The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the entire several recorded states.

The current harvested energy state probabilities results have shown that roughly 61% was the transition from low to high and from high to high was approximately 58% during the simulation time. The threshold of 0.003 J was chosen as the low state of the current harvested energy. If the quantity of the harvested energy is higher than 0.003 J, it is then deemed high. The harvested energy quantities of 0.00897493 J and 0.000102277 J were maximum and minimum harvested energies, respectively.

Figure 3.20 depicts the current harvested energy values during the 600 s duration of the simulation. The energy storage is recharged if the thermoelectric generator state is high. It can be observed that 0.0089749 J and

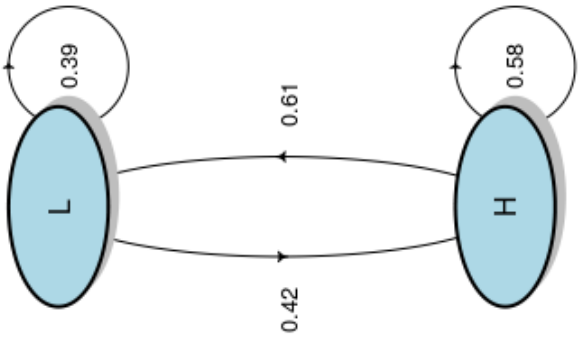

Figure 3.19. State transition probabilities of the harvested energy

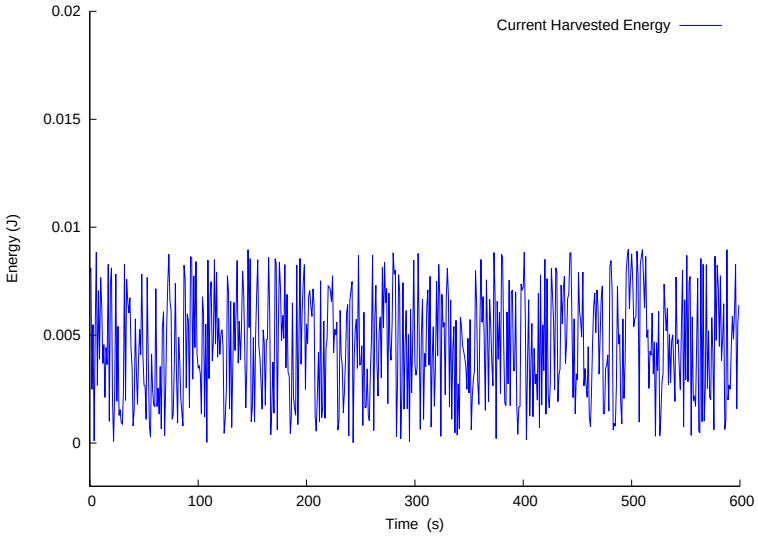

Figure 3.20. Current harvested energy

0.000102277 J were the maximum and minimum, respectively. These harvested energy quantities were reached during the simulation time at each time interval. It can be noted that if the threshold harvested energy of less than 0.0033 J is reached, then no recharging activities would take place.

The receiving and transmission data queues play a vital role in the WSN nodes. It is because they store data before any transmission and before processing when receiving. Data queues can be congested and result in data loss if the WSN energy storage is at a low state. The transceiver is put to sleep if the WSN energy storage is at the low state. Hence, data loss probabilities and the queue length prediction is vital in order to prevent data loss in WSN nodes.

Table 3.8. State transition probabilities of the queue size

	Low	High
Low	0.96	0.04
High	0.05	0.95

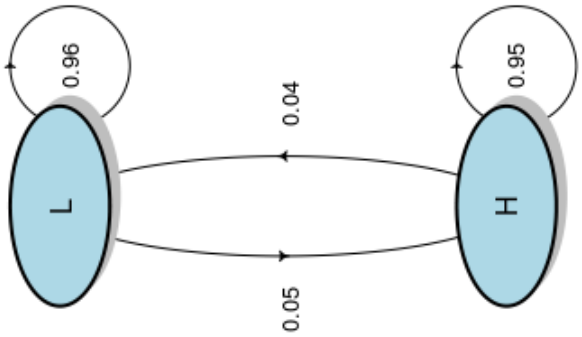

Figure 3.21. State transition probabilities of the queue size

Table 3.8 and Figure 3.21 show the transition probabilities of the queue states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states.

The current state probabilities results of the data queue have shown that the High-to-High transition status is at roughly 95%. It can be observed that around 99% of the simulation time, the data queue status is in High state. This is because, from the simulation parameters, the number of 300 packets was the threshold for the maximum queue capacity, and 200 number of packets was set for the high queue state.

The current queue size during the simulation time is illustrated in Figure 3.22, where the number of 300 packets is pre-set as the maximum queue capacity. If the queue is full, then the packets will be lost. It can be observed that the number of 287 packets was the average queue size. The minimum queue size was 2 packets.

In the simulation, 20135 was the number of packets that were expected to

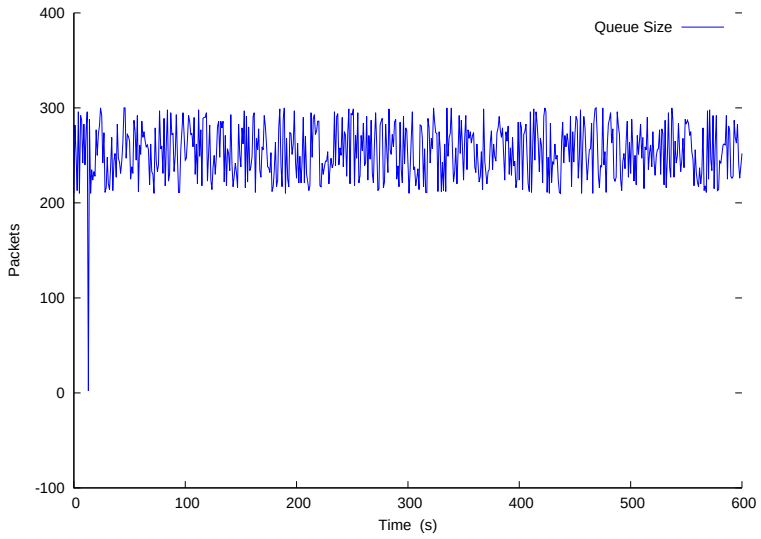

Figure 3.22. Queue Size

arrive at the queue for transmission in the duration of the simulation. The successful number of transmitted packets were 9999. However, 9873 number of packets were dropped, and only 263 number of packets remained in the WSN node queue when the simulation completed. This roughly 49% of the packets were lost. Since the data queue was overflowing when the transceiver was put to sleep due to the low energy storage status, high packet loss probability occurred.

3.6 Integrated PEG and TEG

3.6.1 Harvested Energy

The integrated approach for PEG and TEG is implemented based on the proposed stochastic scheduling presented in Section 3.3. The energy harvested during the simulation time was in high to high transition state at around 93% and in low to high transition state at approximately 99%. These results are shown in Table 3.9 and Figure 3.23. The results have shown to be preferable than if PEG and TEG were simulated separately. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states.

Table 3.9. State transition probabilities of harvested energy

	Low	High
Low	0.01	0.99
High	0.07	0.93

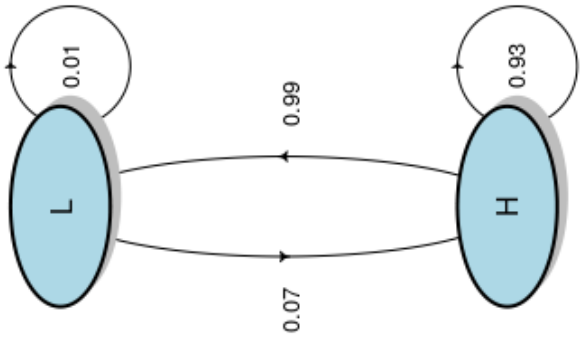

Figure 3.23. Harvested energy state transition probabilities

Figure 3.24 depicts the current harvested energy values during the 600 s duration of the simulation. It can be observed that 0.0099805 J and 0.0030182 J were the maximum and minimum, respectively. These harvested energy quantities were reached during the simulation time at each time interval.
The comparisons between the individual PEG, TEG and the integrated TEG and PEG is illustrated in Figure 3.25. The harvested energy in the integrated

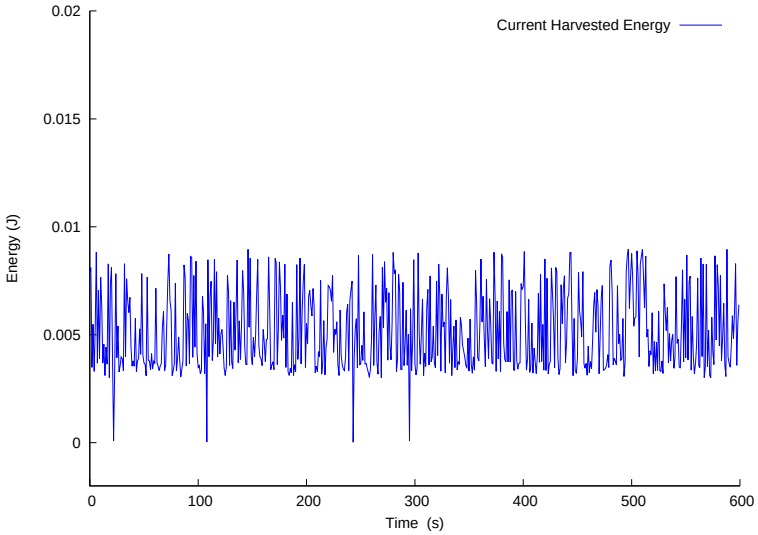

Figure 3.24. Current harvested energy

TEG and PEG approach was around 35% more energy than the individual TEG. The harvested energy in the integrated TEG and PEG approach was also found to be approximately 70% more energy than the individual PEG.

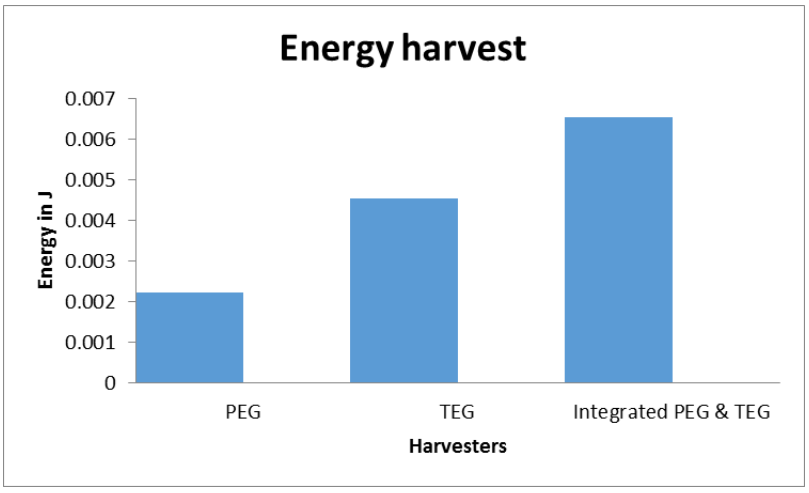

Figure 3.25. Average harvested energy

3.6.2 Energy Storage

The energy storage values of the integrated PEG and TEG approach is shown in Figure 3.26. The results of the integrated PEG and TEG approach have shown to have better energy storage values compared to individual TEG and PEG.

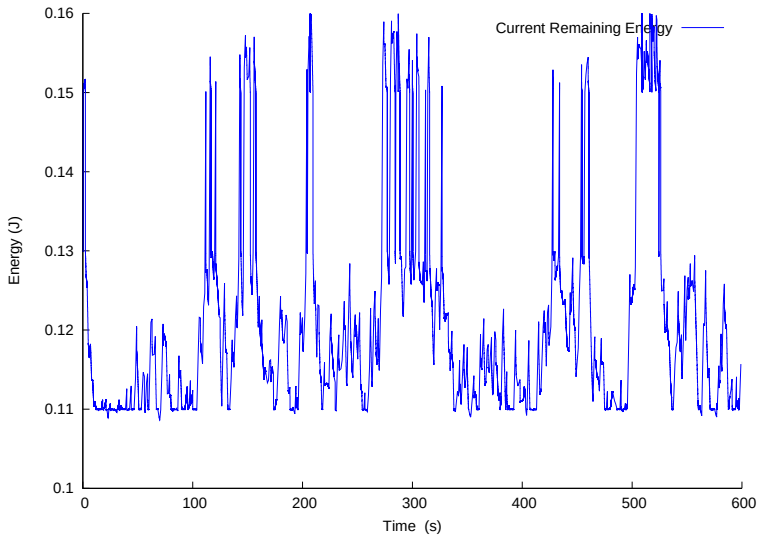

Figure 3.26. Current energy storage values

Table 3.10 and Figure 3.27 depict the transition probabilities of the current energy storage states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states.

The current energy storage state probabilities results have shown that the High to High transition status is at roughly 86% and from low to high transition state was at around 74% of the simulation time. It is because, from the simulation parameters, 0.11 J was the threshold for the low current energy. Otherwise, if the current energy quantity is higher than 0.11 J, it is deemed high. 0.15J was set as the threshold for the maximum energy capacity of the storage. The range of these values were the only possible values for the energy storage model in NS3. The transceiver is put to sleep in order to save energy if the energy storage status is Low.
The integrated PEG and TEG approach is compared to individual TEG and PEG in Figure refstorage-comp. It can be observed that more average energy storage is obtained in the integrated approach compared to the indi-

Table 3.10. State transition probabilities of energy storage

	Low	High
Low	0.26	0.74
High	0.14	0.86

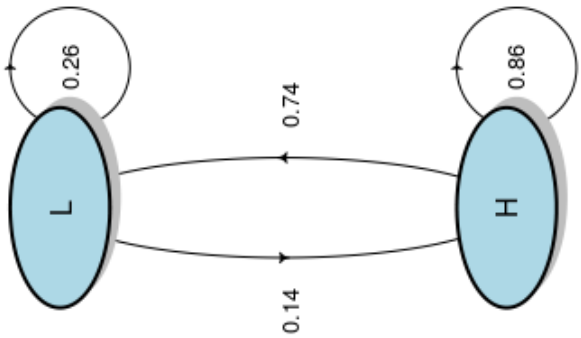

Figure 3.27. State transition probabilities of energy storage

vidual PEG and TEG approach. The integrated approach has around 3% more energy stored compared to TEG and around 6% more energy stored compared to PEG.

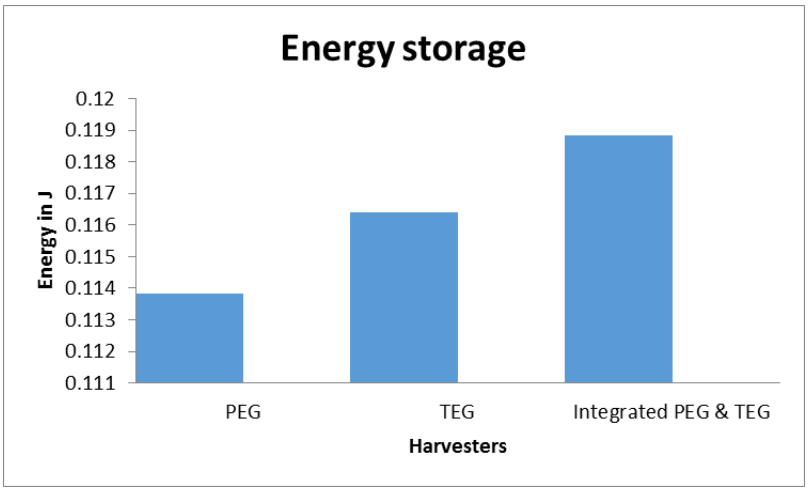

Figure 3.28. Average energy storage

The integrated approach has around 3% more energy stored compared to TEG and around 6% more energy stored compared to PEG.

3.6.3 Packet Loss Probability

In the simulation, 26459 was the number of packets that were expected to arrive at the queue for transmission in the duration of the simulation. The successful number of transmitted packets were 20199. However, 6201 number of packets were dropped, and only 59 number of packets remained in the WSN node queue when the simulation completed. This roughly 23% of the packets were lost.

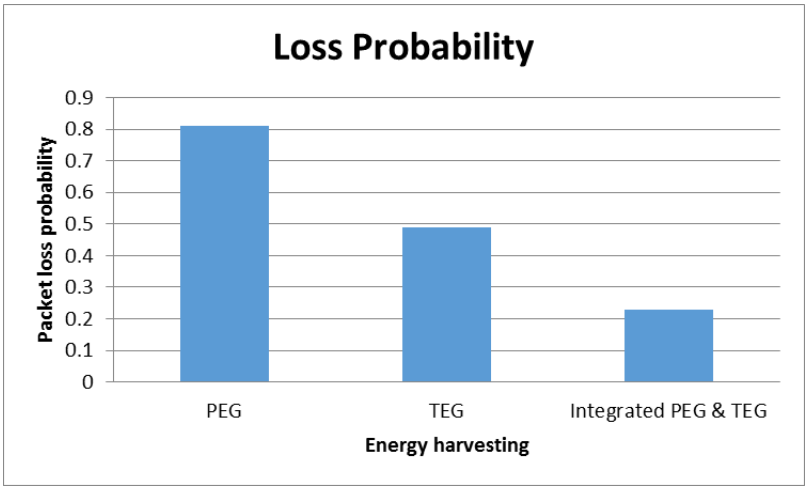

Figure 3.29. Packet loss probability

It can be observed in Figure 3.29 that packet loss probability in the integrating PEG and TEG was lower than the individual TEG and PE. This is because high energy harvesting is experienced in the integrated PEG and TEG approach due to scheduling technique and therefore, the number of transceiver sleep modes less frequent.

The deployment of the integrated PEG and TEG with effective scheduling scheme will yield high energy harvesting less packet loss. The loss probability was lower compared to the individual deployment of PEG and TEG. This because the gap of harvesting inactivity was reduced by the integrated approach. The integrated approach had 23% packet loss while TEG had around 49%, and PEG had almost 80%. Though the simulation of the integrated approach has shown to have less packet loss compared to others, implementing it in practice can be challenging. This is because placing both harvesters in close proximity on the aircraft could be a challenge.

3.7 Extended Energy Harvesting, Storage and Data Queue States

This section will present the extended states for energy harvesting, data queue and energy storage. The preceding section had used only two states (Low and High). Two states do not capture the real-world scenario. Five states which are Low, Low Medium, Medium, High Medium and High closely represent the real-world scenario.

The states are extended to capture the near reality of the real world states and improve the results. The states for harvested energy, storage and data queue states are extended to, High (H), High Medium (HM), Medium (M), Low Medium (LM) and Low (L).

Each harvested energy state, storage state and data queue state has its predetermined threshold. The same notations are used as in Section 3.2 whereby (p,t,s,d) denotes the state of the WSN node where the harvesting state for PEG, TEG, the energy source state and the data queue is $p,t,s,d,$ respectively. The steady-state probability for the state (p,t,s,d) is represented by $\pi(p,t,s,d)$. The following notations are also used,

- p_h, p_{hm}, p_m, p_{lm} and p_l are the harvested energy state for PEG is high, high medium, medium, low medium and low, respectively.
- t_h, t_{hm}, t_m, t_{lm} and t_l are the harvested energy state for TEG is high, high medium, medium, low medium and low, respectively.
- s_h, s_{hm}, s_m, s_{lm} and s_l are the energy storage state is high, high medium, medium, low medium and low, respectively.
- d_h, d_{hm}, d_m, d_{lm} and d_l are the data queue state for a node is high, high medium, medium, low medium and low, respectively.

3.7.1 Stochastic Scheduling

The proposed Markov base model for energy harvesting, the energy storage capacity, packets queue length and WSN power consumption are used by the stochastic scheduling scheme to make decisions on which harvesting source should be picked up to charge or recharge the energy storage.

Case 1: If p_h and t_h and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 2: If p_h and t_h and s_h and $d_h m$ at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 3: If p_h and t_h and s_h and d_m at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 4: If p_h and t_h and s_h and $d_l m$ at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 5: If p_h and t_h and s_h and d_l at time slot t_i then put the transceiver to sleep and stop energy harvesting.
Case 6: If p_h and t_h and $s_h m$ and d_h at time slot t_i then transmit the data from the data queue and start energy harvesting either from PEG or TEG depending on which source has higher energy
Case 7: If p_h and t_h and s_m and d_h at time slot t_i then transmit the data from the data queue and start energy harvesting either from PEG or TEG depending on which source has higher energy
Case 8: If p_h and t_h and $s_l m$ and d_h at time slot t_i then transmit the data from the data queue and start energy harvesting either from PEG or TEG depending on which source has higher energy
Case 9: If p_h and t_h and s_l and d_h at time slot t_i then put the transceiver to sleep and start energy harvesting either from PEG or TEG depending on which source has higher energy
Case 10: If p_h and $t_h m$ and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 11: If p_h and t_m and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 12: If p_h and $t_l h$ and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 13: If p_h and t_l and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 14: If $p_h m$ and t_h and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 15: If p_m and t_h and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 16: If $p_l h$ and t_h and s_h and d_h at time slot t_i then transmit the data from the data queue and stop energy harvesting
Case 17: If p_l and t_h and s_h and d_h at time slot t_i then transmit the data from

the data queue and stop energy harvesting
There are 625 cases in total. Due to space limitation, this thesis has only listed 17 cases. One of the above described cases will be observed at any time slot t_i in the simulation.

3.7.2 Simulation Setup for Integrated PEG and TEG

This section also uses NS3 simulation as in Section 3.4. The sample code is appended in Appendix A.

The simulation parameters are configured to depict the distribution of WSN nodes on the aircraft and the total number of packets sent in an interval of 20ms.

- The maximum number of packets to send is set to 100000 packets
- The arrivals or departures of the packets are set 20ms
- The total WSN nodes number is set to 150
- 20ms is chosen as the time slot
- The simulation is started at time 0 s
- The simulation duration is set to 600 s
- Deployment area = $100x100m^2$
- Nodes deployment = random
- 20 m is set as the minimum distance between nodes
- 0.11J is set as the threshold battery level for recharging the storage capacity
- 0.15 J is set as the threshold for maximum battery level capacity
- The IEEE protocol for wireless communication is chosen as 802.15.4 with transmission and receiving power as TX = 0.0174J and RX = 0.0197J, respectively
- he piezoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.0022 J - 0.0044 J]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [34, 35].

- The thermoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.003 0.22]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [36].

3.7.3 Simulation Results for Integrated PEG and TEG

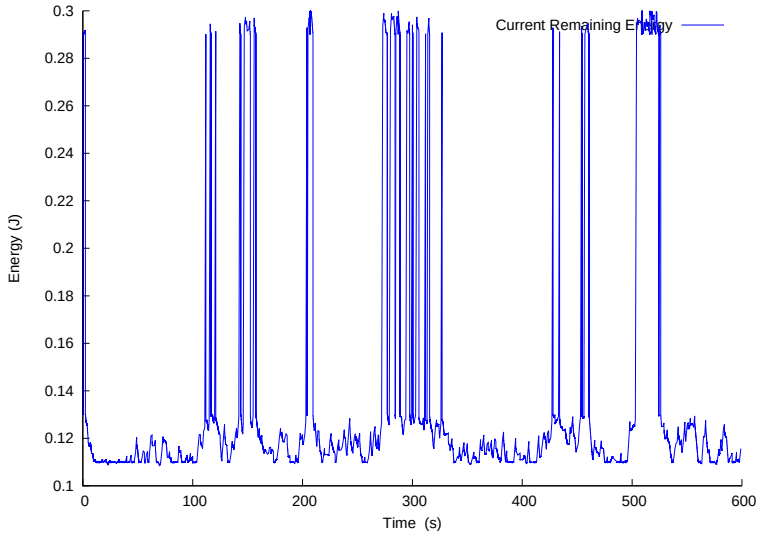

Figure 3.30. Current energy storage values

The results of the integrated PEG and TEG harvesting approach is depicted in Figure 3.30 with more Markov states. It shows that an integrated PEG and TEG approach with more Markov states results in preferable energy storage values. These results are caused by the integrated PEG and TEG approach with five Markov states harvested more energy compared to only two states. The minimum storage value was 0.11J, and the maximum value was 0.29J. This is about a 64% increase in energy storage with five Markov states compared to only two.

Table 3.11 and Figure 3.31 show the transition probabilities of the current energy storage states. The transition probabilities are approximated by conducting many simulation runs. Transition state probabilities are obtained during simulation by recording the ratio of how many times a state has occurred over the total number of recorded states. The transition from other states to high dominated the transition states as shown in the results from the current energy storage states probabilities
The comparison between integrated PEG and TEG approaches with two

Table 3.11. Energy storage state transition probabilities

	L	LH	M	HM	H
L	0.1200	0.1000	0.0400	0.3000	0.4400
LH	0.1000	0.1400	0.3000	0.1000	0.3600
M	0.2000	0.1000	0.1600	0.0800	0.4600
HM	0.0500	0.1000	0.0500	0.1900	0.6100
H	0.0200	0.1300	0.1500	0.2000	0.5000

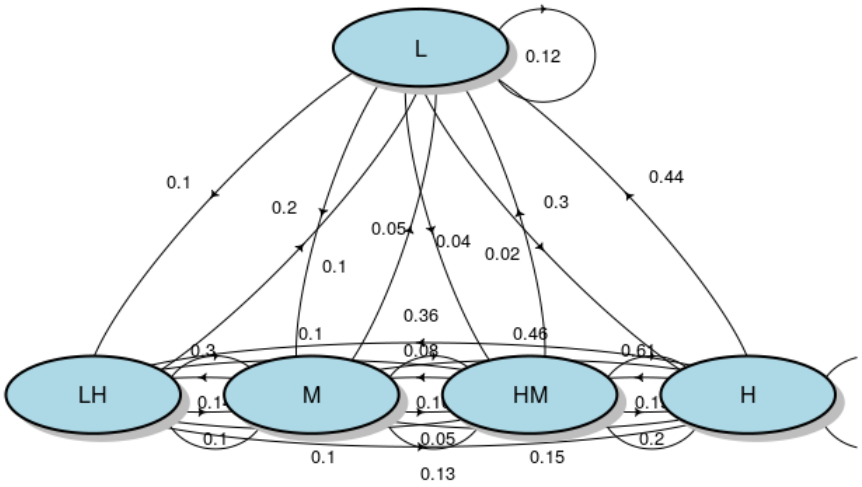

Figure 3.31. Energy storage states transition

and five Markov states and individuals TEG and PEG are shown in Figure 3.32. The integrated approach with more states transition is shown to have more average energy storage compared to only two states transitions. There was an increase of about 64% in energy storage with five Markov states compared to only two Markov states.

By integrating PEG and TEG with more Markov states, It can be observed that there the packet loss probability lower than the integrated TEG and PEG approach with only two states (Figure 3.33). The packet loss probability of the integrated approach with five Markov states was found to be 2% compared to that of two Markov states which were found to be 23%. The stochastic scheduling with five Markov states obtained higher harvested en-

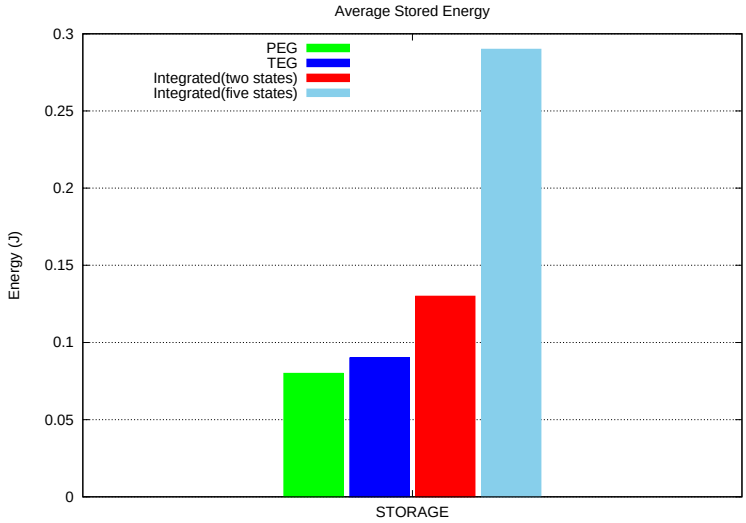

Figure 3.32. Average energy storage

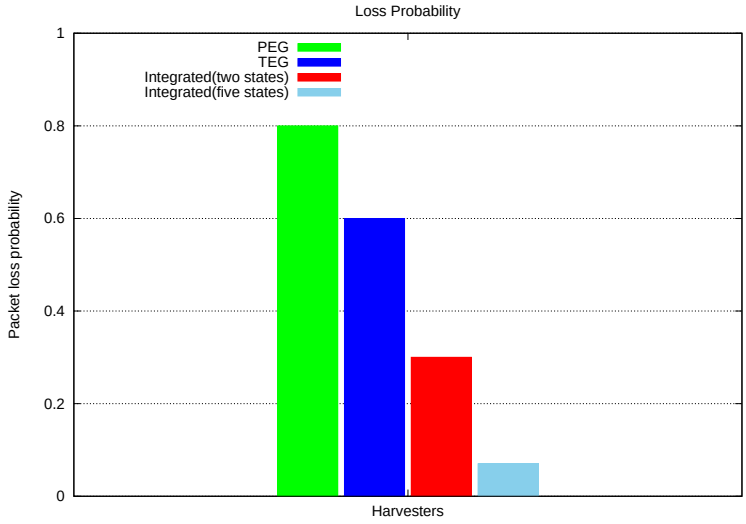

Figure 3.33. Packet loss probability

ergy and therefore, higher energy storage compared to two Markov states. These results will cause to eventually have less radio channel sleep duration. The results have also shown that the integrated TEG and PEG with five states has resulted in more harvested energy (c.f., Figure 3.34).

The comparison between the integrated PEG and TEG with five Markov states, the integrated PEG and TEG with two Markov states and individuals TEG and PEG are depicted in Figure 3.34. The integrated PEG and TEG approach with five Markov states showed that it harvested more energy than the rest (about 64% more energy than the one with two Markov states).

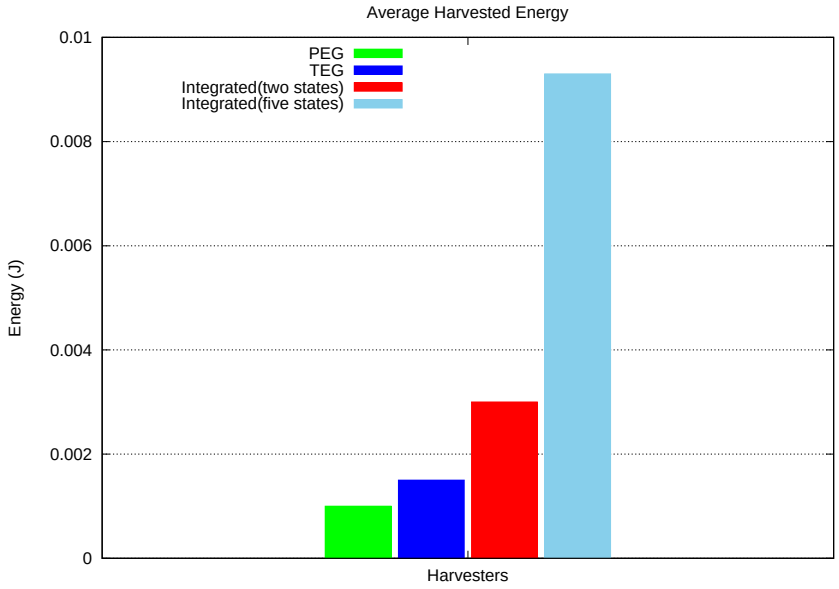

Figure 3.34. Average harvested energy

The integrated PEG and TEG approach with five Marko states with efficient scheduling scheme obtained higher harvested energy which caused less packet losses. The empirical results demonstrated that there was a minimal increase (of just 1%) in energy harvesting at the expense of more computation time (an increase of around 30%).

3.8 Summary

The novel idea of the Integrated PEG and TEG approach was presented in this chapter. Two Markov states (Low and High) for energy harvesting, storage and data queue were initially used. The two Markov states did not capture the real-world scenario because there are some states between Low and High. Five Markov states (Low, Low Medium, Medium, High Medium and High) were deployed that were closely represent the real world scenario instead of two Markov states.

The integrated PEG and TEG approach with two Markov states were further extended to include 5 states transition instead of only two (Low and High), this depicts the more realistic scenario. The integrated PEG and TEG approach with five Markov states, together with stochastic scheduling, was shown to get fewer packet losses. These results are caused by the gap of inactivity of energy harvesting is reduced compared to when individual TEG and PEG harvesters.

More average energy storage was obtained in the integrated approach with five Markov states compared to only two states transitions. There was an increase of about 64% in energy storage with five Markov states compared to only two Markov states.

The packet loss probability of the integrated PEG and TEG approach with five Markov states was found to be 2% compared to that of two Markov states which were found to be 23%. These results were caused by the integrated PEG and TEG approach together with effective stochastic scheduling with five Markov states obtained higher harvested. Higher harvested energy meant high energy storage and less frequent transceiver sleep modes.

The results also showed that the integrated approach with five Markov states showed that it harvested more energy than the rest. It was about 64% more harvested energy than the one with two Markov states.

4 LTspice and NS3 INTEGRATION

4.1 Overview

This thesis has found that there is a gap of integration between the popular circuit simulation package LTspice and the network simulation package NS3. In this respect, this chapter aims at integrating the two simulators LTspice and NS3. This integration will cement robust research cooperation amongst electronics and networking research domains. The LTspice and NS3 integration has been validated by deploying the Fuzzy logic control approach power management use case in energy harvesting.
The LTspice is widely used by researchers in electronics to simulate electronics circuits. The NS3 is widely used by communication and networking researcher. They are both freely available. If the integration of LTspice and NS3 is successful, then the power output from LTspice will be used by the NS3 network modules. The proposed system in Chapter 3 used a Markov chain model for energy harvesting. Markov based schemes are prediction-based, and therefore, the quantity of energy the WSN node can harvest will demonstrate energy variations. This will make it difficult to predict. As a result, energy harvesting prediction models are prone to faults. The harvested energy would either be over-use or under-use and, therefore, model-free prediction models are preferred for power management strategies. Hence, this study proposes a Fuzzy control logic expert system for power management.

4.2 Introduction

Current development tools need to be able to integrate Wireless Sensor Networks increasingly in industries and critical infrastructures. Before deployment, either on-site experiments or simulations need to evaluate the performance of the WSN. The performance evaluation on-site consumes time and is expensive. Therefore, the full range of predicted operating conditions can not quickly be addressed and can not be replicated with real sensor nodes. Therefore, simulations are the only efficient way to assess wireless network performance. The fact that energy is required for network elements is a significant problem associated with most current network simulation models is because energy models vary from those implemented in actual devices. It is therefore often difficult to decide whether these performance characteristics

are based on particular protocol features or are just a feature of a certain network simulator implementation. In this context, energy source models from circuit simulation tools such as LTspice can be used to supply realistic energy models to the network simulation packages.

Consequently, the integration of these simulators complements one another. The network communication protocols that are commonly available in NS3 can allow LTspice to take advantage of them while NS3 will take advantage of simulated integrated circuit nodes from LTspice. In this thesis, NS3 and LTspice integration is focused on the power energy harvesting power management scenario.

To demonstrate the feasibility of the LTspice and NS3 integration, this thesis presents the implementation of the energy harvesting power management system by using piezoelectric energy generators from the LTspice. The power generated by the PEG in LTspice is then communicated to NS3 by using a TCP channel. LTspice does not have TEG modules, therefore, only PEG was implemented.

4.3 The Design of the Proposed Integration

The Nanopower Energy Harvesting Power Supply LTC3588-1 module [124] is used in this thesis. The high-efficiency buck converter can be integrated by the LTC3588-1 with a full wave low loss bridge rectifier. This is done to formulate a solution for full energy harvesting optimized for impedance energy sources with high output such as solar, piezoelectric, thermoelectric and magnetic transducers.

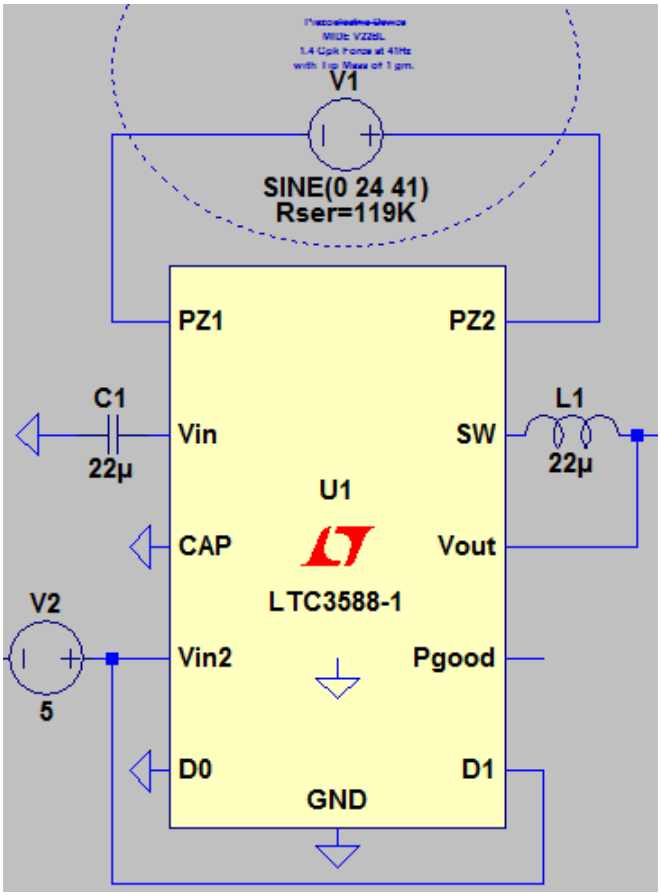

Figure 4.35. LTC3588-1 - power supply

The four output voltages which can be selected are 3.6 V, 3.3 V, 2.5 V and 1.8 V. The associated current can reach current [154]. LTC3588-1 power supply with PEG node is shown in Figure 4.35. The voltages, V1 which is in the form of the Sinewave for the PEG and V2 in the form of Pulse to control signals of the analogue to digital enable command.

The TCP channel encapsulates V1 and V2 from LTspice to the NS3 simulation via the lightweight virtual communication container (c.f., Figure 4.36). The simulation parameters are configured as in the previous chapter to de-

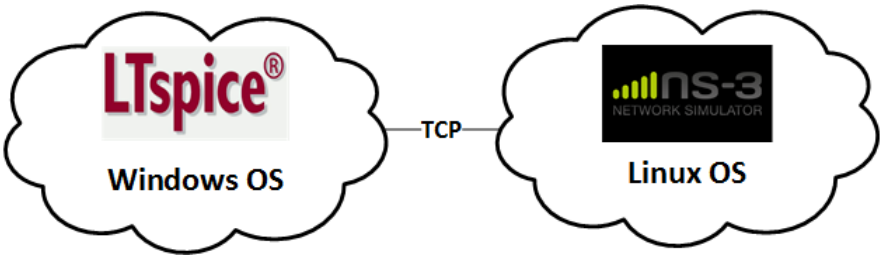

Figure 4.36. LTspice and NS3 TCP connection

pict the distribution of WSN nodes on the aircraft and the total number of packets sent in an interval of 20ms. Note that these values are based on the literature review presented in Section 2.1.1 [34, 35]. NS3 simulation parameters were as follows,

- The maximum number of packets to send is set to 100000 packets
- The arrivals or departures of the packets are set 20ms
- The total WSN nodes number is set to 2
- 20ms is chosen as the time slot
- The simulation is started at time 0 s
- The simulation duration is set to 600 s
- 20 m is set as the minimum distance between nodes
- 0.11J is set as the threshold battery level for recharging the storage capacity
- 0.15 J is set as the threshold for maximum battery level capacity
- The IEEE protocol for wireless communication is chosen as 802.15.4 with transmission and receiving power as TX = 0.0174J and RX = 0.0197J, respectively
- The piezoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.0022 J - 0.0044 J]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [34, 35].
- These power values are supplied by the Nanopower Energy Harvesting Power Supply LTC3588-1 [154] with the input operating range of 5V at Vin

4.4 Implementation of the Design

The Fuzzy control expert system is used for the implementation of the proposed LTspice and NS3 integration. The energy values are generated and transmitted from LTSpice to NS3. The Fuzzy control is used in the integration as an expert based system to intelligently make decisions based on IF-THEN statement. These decisions include making WSN nodes to go to sleep mode or making it active, harvesting and not harvesting and charging and stopping charging the battery.

The proposed fuzzy control expert system design showing the Residual Energy (RE) and Harvested Energy (HE) as inputs, output actions and inference engine is shown in Figure 4.37. The fuzzy inference engine formulates the mapping from a given input (HE and RE) to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made (actions). These actions are sleep and active.

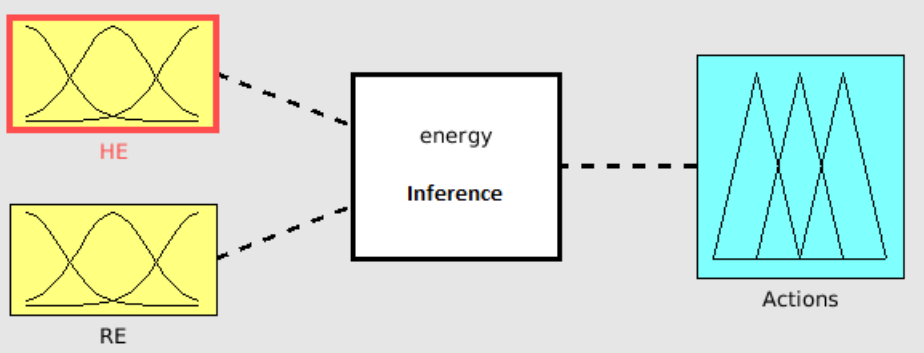

Figure 4.37. Fuzzy control system design

4.4.1 Fuzzification Process

The following inputs are considered at any time slot t_i of the simulation,

- **Harvested Energy (HE)**
  - The LOW and HIGH are the two harvested energy fuzzy crisp sets
  - The lowest amount of harvested energy is denoted by LHE is required to run the WSN
  - The threshold for harvested energy to be considered as HIGH is denoted by MHE

- The membership functions which are normalized are,
  - If HE ≤ LHE the LOW = 1
  - If HE ≥ MHE then LOW = 0
  - If LHE < HE < MHE then LOW = x

  - If HE ≥ MHE then HIGH = 1
  - IFHE ≤ LHE then HIGH = 0
  - If LHE < HE < MHE then HIGH = x

The membership function for the harvested energy is shown in Figure 4.38.

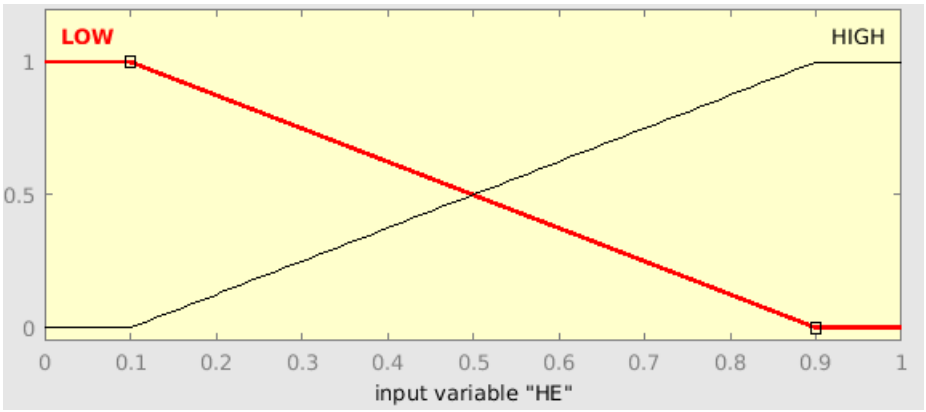

Figure 4.38. Membership function of harvested energy

- **Residual Energy (RE)**
  - EMPTY and FULL are the two residual energy fuzzy crisp sets
  - The amount of energy required is denoted as ERE is reserved for powering the WSN node
  - The sufficient reserved RE by the WSN is denoted as FRE
  - The membership functions are normalized and shown below,
    - If RE ≥ FRE then FULL=1
    - If RE ≤ ERE then FULL = 0 and
    - If ERE < RE < FRE then FULL=x

    - If RE ≤ ERE then EMPTY=1
    - If RE ≥ FRE then EMPTY=0 and

* If ERE < RE < FRE then EMPTY = x

The membership function of the residual energy is shown in Figure 4.39.

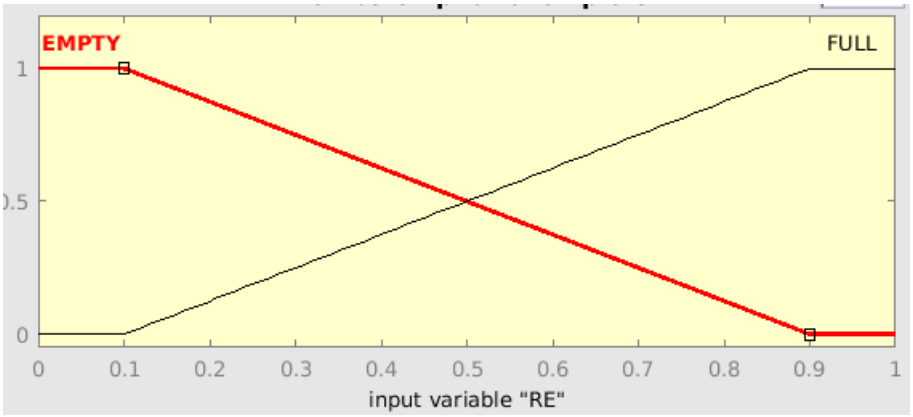

Figure 4.39. Membership function of the residual energy

4.4.2 Inference engine

A set of 4 fuzzy IF-THEN rules describe the interference engine and is shown in Figure 4.40. Fuzzy rules are used within fuzzy logic systems to infer an output (sleep or active) based on input variables (HE or RE).

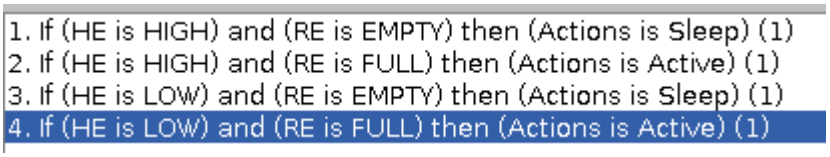

Figure 4.40. IF-THEN Fuzzy rules

The following rules are addressed in this thesis at any time slot t_i of the simulation,

- Rule 1 (R1): If the state of the energy storage is EMPTY and the state of the harvested energy is HIGH then let part of the energy harvested to be used to power the WSN node, and part of it will be stored.
- Rule 2 (R2): If the state of the energy storage is FULL and the state of the harvested energy is HIGH, then let part of the harvested energy to be used to power the WSN node.
- Rule 3 (R3): If the state of the energy storage is EMPTY and the state of the harvested energy is LOW then let the harvested energy to be stored and the WSN transceiver to be put to the sleep mode.

- Rule 4 (R4): If the state of the energy storage is FULL and the state of the harvested energy is LOW, then let the stored energy be used to power the WSN node.

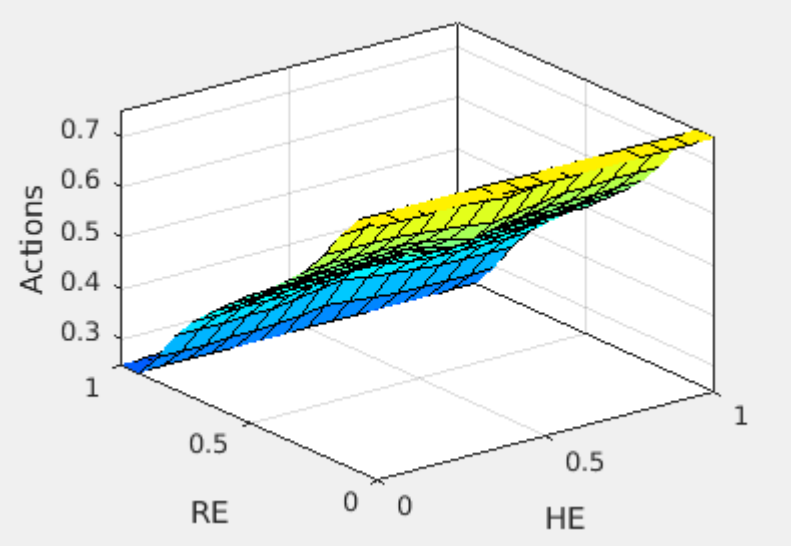

Figure 4.41. Surface rules view

4.4.3 Defuzzification

The defuzzification provides the inference engine output. These are the energy output due to the action of the fuzzy rules. Figure 4.41 shows the 3-dimensional view of the fuzzy inference rules. The figure clearly shows all three parameters in one graph, which are normalized residual, harvested and actions taken by the system. The actions taken by the systems are based on the fuzzy rules such as power the WSN, store the energy, wake up and put the transceiver to sleep.

4.5 Simulation Results

4.5.1 Evaluation metrics

The Energy Utilization Efficiency (EUE) and the Downtime Ratio (DR) are used as evaluation metrics and before presenting simulation results.

- Downtime ratio (DR)
  - The Downtime Ratio is defined as the ratio the WSN node spend on the sleep mode to the simulation time. This ratio is in the range of 0 to 1. If the Downtime Ration 0, this means that the WSN transceiver was never put in the sleep mode. If the Downtime Ratio is 1, this means that during the simulation time, the WSN transceiver was put in the sleep mode all the time.
- Energy Utilization Efficiency (EUE)
  - Energy Utilization Efficiency is defined as the ratio between the harvested energy used by the WSN node and the total energy harvested during the simulation time. The EUE is in the range of 0 to 1 included. If the EUE is 0, this means that the WSN node never used the harvested energy during the simulation time. If the EUE is 1, this means that during the simulation time, the WSN node was continuously using the harvested energy.

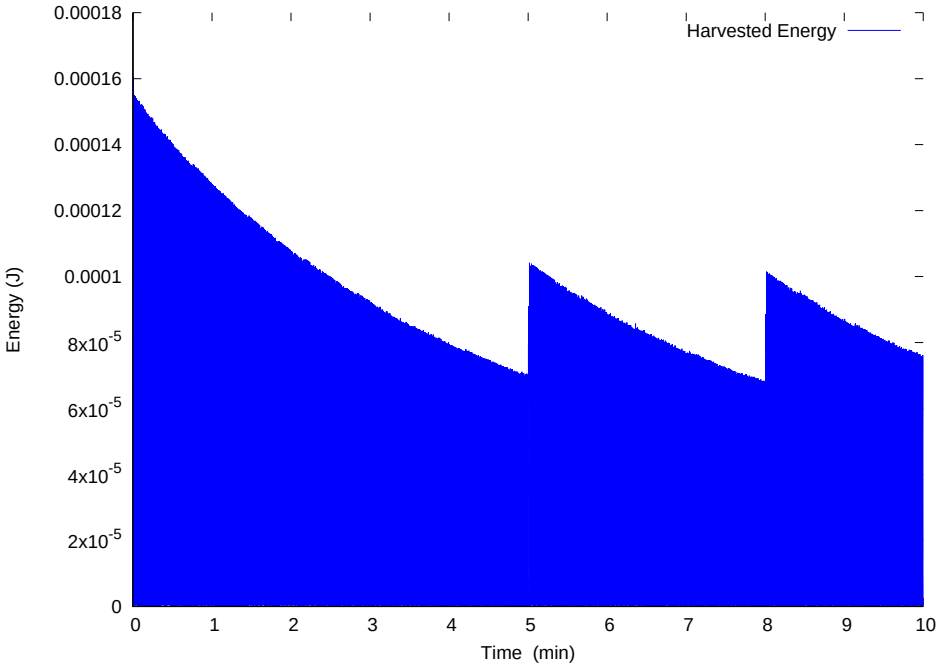

Figure 4.42. Harvested energy

During the 10 minutes of the simulation time, the harvested energy from LTspice is depicted in Figure 4.42. The energy quantities of 1.57e-004 J and 9.00e-05 J were the maximum and minimum harvested energies, respectively. The average harvested energy was 4.74617e-05 J. These energy values based on the values presented in Chapter 2 [34, 35].

During the 10 minutes of the simulation time, the residual energy values are depicted in Figure 4.43. The residual energy quantities of 1.0 J and 6.003e-08 J were the maximum and minimum residual energies, respectively. The average harvested energy was 0.63 J.

The simulation results showed that the Downtime ratio was 0.3233. Due to fuzzy control system deployment, roughly 9% of the simulation time, the WSN transceiver was put to sleep. For efficient WSN node communication, the DR close to zero is the ideal solution.

Figure 4.44 depicts the quantity of energy used by the WSN node. Energy Utilization Efficiency was observed to be 85%. The EUE close to 100% would be the ideal solution because that would show that the WSN node used all the harvested energy.

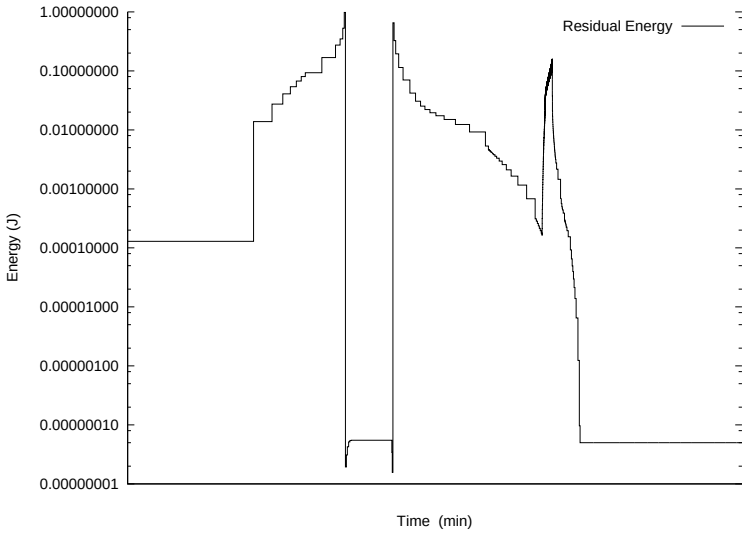

Figure 4.43. Residual energy

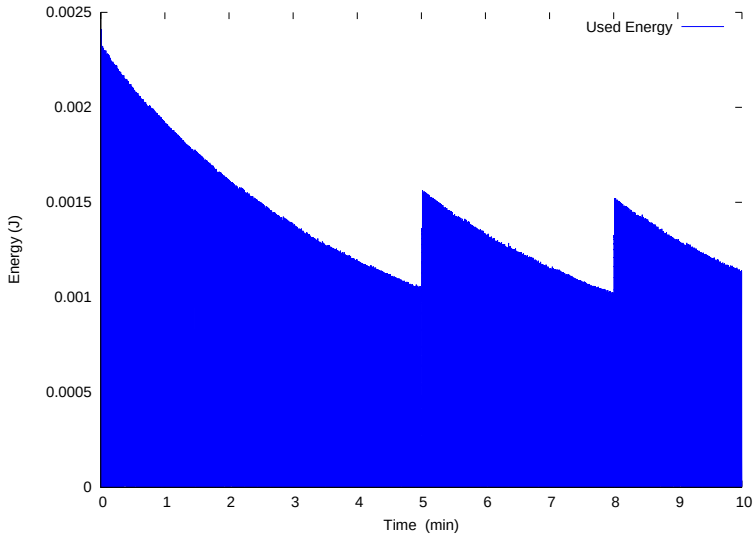

Figure 4.44. Used energy

4.5.2 Fuzzy approach Vs non-fuzzy approach

Figure 4.45 depicts the results of the comparison between fuzzy and non-fuzzy approaches. It was observed that the values of the DR and the EUE for the fuzzy-based better compared to the ones without fuzzy control deployment.

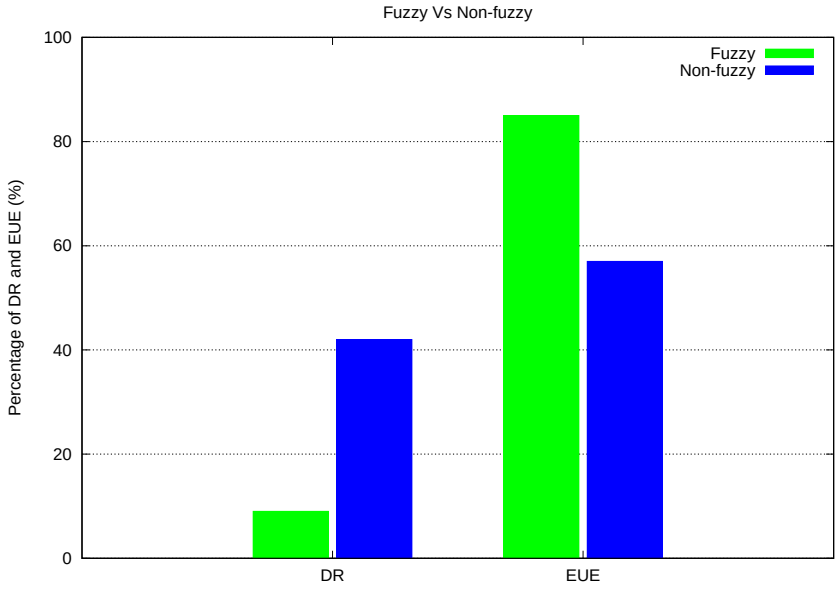

Figure 4.45. Fuzzy vs Non-Fuzzy approaches

4.6 Extended Fuzzy Sets

The fuzzy sets are extended to capture the near reality of the real world states and improve the results. The fuzzy sets for harvested energy are extended to High (H), High Medium (HM), Medium (M), Low Medium (LM) and Low (L).

The fuzzy sets for residual energy are extended to Full (F), HALF FULL (HF), Medium (M), HALF EMPTY (HE) and EMPTY (E),
The following inputs are considered at any time slot t_i of the simulation.

- **Harvested Energy (HE)**
  - HHE denotes the High state of the harvested energy threshold
  - HMHE denotes the HIGH MEDIUM state of the harvested energy threshold
  - MHE denotes the MEDIUM state of the harvested energy threshold
  - LMHE denotes the LOW MEDIUM state of the harvested energy threshold
  - LHE denotes the LOW state of the harvested energy threshold
  - The membership functions are normalized and listed below,

* If HE ≤ LHE then LOW =1
* If HE ≥ LHE then LOW = 0 and
* If LHE < HE < LMHE then LOW = x

* If HE ≤ LMHE then LOW MEDIUM =1
* If HE ≥ LMHE then LOW MEDIUM =0 and
* If LHE < HE < LMHE then LOW MEDIUM = x

* If HE ≤ MHE then MEDIUM = 1
* If HE ≥ MHE then MEDIUM = 0 and
* If LMHE < HE < MHE then MEDIUM = x

* If HE ≤ HMHE then HIGH MEDIUM = 1
* If HE ≥ HMHE then HIGH MEDIUM = 0 and
* If MHE < HE < HMHE then HIGH MEDIUM = x

* If HE ≥ HHE then HIGH =1
* If HE ≤ HHE then HIGH = 0 and
* If HMHE < HE < HHE then HIGH = x

The harvested energy membership function is depicted in Figure 4.46.

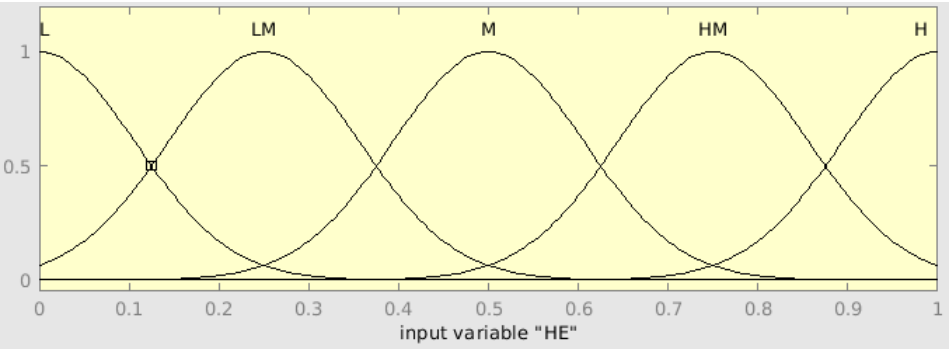

Figure 4.46. Membership function of the harvested energy

- **Residual Energy (RE)**
  - To run the WSN node, ERE is the quantity of energy to be reserved
  - HERE is the amount of energy that is considered HALF EMPTY
  - MERE is the amount of energy that is considered MEDIUM
  - HFERE is the amount of energy that is considered HALF FULL
  - WSN node is deemed to reserve sufficient RE denoted by the FRE
  - Membership functions are listed below and are normalized,
    - If RE $\leq$ ERE then EMPTY =1
    - If RE $\geq$ ERE then EMPTY = 0 and
    - If ERE < RE < HERE then EMPTY = x
    - If RE $\leq$ HERE then HALF EMPTY = 1
    - If RE $\geq$ HERE then HALF EMPTY = 0 and
    - If ERE < RE < HERE then HALF EMPTY = 0
    - If RE $\leq$ MRE then MEDIUM = 1
    - If RE $\geq$ MRE then MEDIUM = 0 and
    - If HERE < RE < MRE then MEDIUM = x
    - If RE $\leq$ HFRE then HALF FULL = 1
    - If RE $\geq$ HFRE then HALF FULL = 0 and
    - If MRE < RE < HFRE then HALF FULL = x
    - If RE $\leq$ FRE then FULL = 1
    - If RE $\geq$ FRE then FULL = 0 and
    - If HFRE < RE < FRE then FULL = x

The residual energy membership functions are depicted in Figure 4.47.

4.6.1 Inference engine

Figure 4.48 shows the inference engine outlined by IF-THEN set of fuzzy rules. Due to space limitation, only 13 rules are listed out of 625 rules.

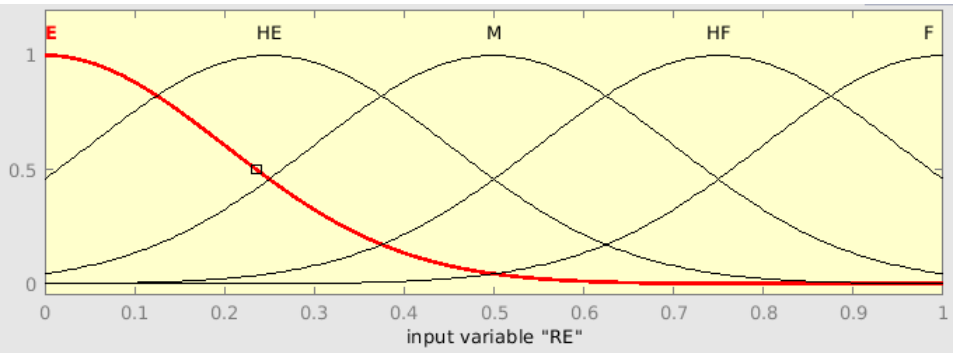

Figure 4.47. Residual energy membership function

1. If (HE is L) and (RE is E) then (Actions is Sleep) (1)
2. If (HE is LM) and (RE is E) then (Actions is Sleep) (1)
3. If (HE is M) and (RE is E) then (Actions is Sleep) (1)
4. If (HE is HM) and (RE is E) then (Actions is Sleep) (1)
5. If (HE is H) and (RE is E) then (Actions is Sleep) (1)
6. If (HE is L) and (RE is E) then (Actions is Charge) (1)
7. If (HE is LM) and (RE is E) then (Actions is Charge) (1)
8. If (HE is M) and (RE is E) then (Actions is Charge) (1)
9. If (HE is HM) and (RE is E) then (Actions is Charge) (1)
10. If (HE is H) and (RE is E) then (Actions is Charge) (1)
11. If (HE is L) and (RE is HE) then (Actions is Charge) (1)
12. If (HE is L) and (RE is M) then (Actions is Charge) (1)
13. If (HE is L) and (RE is HE) then (Actions is Charge) (1)

Figure 4.48. IF-THEN Fuzzy rules

4.6.2 Defuzzification

Figure 4.49 illustrates the 3-Dimension view of the inference rules. The figure clearly shows all three parameters in one graph, which are normalized residual, harvested and actions taken by the system.

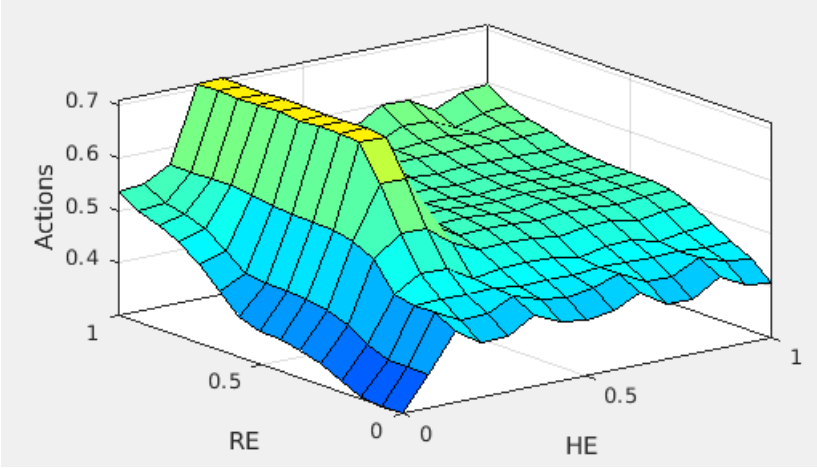

Figure 4.49. Surface rules view

4.7 Simulation Results

The following simulation parameters are configured to depict the distribution of WSN nodes on the aircraft and the total number of packets sent in an interval of 20ms. NS3 simulation parameters are listed below. Note that these values were based on the literature review presented in Section 2.1.1 [34, 35].

- The maximum number of packets to send is set to 100000 packets
- The arrivals or departures of the packets are set 20ms
- The total WSN nodes number is set to 150
- 20ms is chosen as the time slot
- The simulation is started at time 0 s
- The simulation duration is set to 600 s
- Deployment area = 100x100m2
- Nodes deployment = random
- 20 m is set as the minimum distance between nodes
- 0.11J is set as the threshold battery level for recharging the storage capacity
- 0.15 J is set as the threshold for maximum battery level capacity

- The IEEE protocol for wireless communication is chosen as 802.15.4 with transmission and receiving power as TX = 0.0174J and RX = 0.0197J, respectively
- The piezoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.0022 J - 0.0044 J]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [34, 35].

The same metrics outlined in Section 4.5.1 are used, i.e., Downtime ratio and Energy Utilization Efficiency. Average harvested energy, storage energy and packet loss probabilities will then be compared to those of Markov chain model presented in Chapter 3.

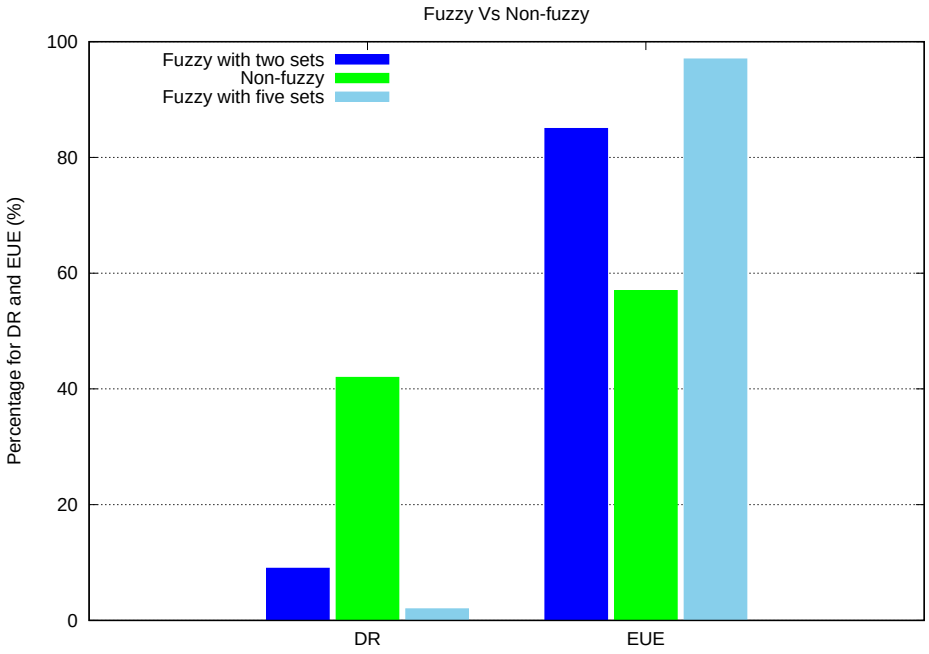

Figure 4.50. Fuzzy Vs Non-Fuzzy approaches

4.7.1 Fuzzy approach Vs non-fuzzy approach

The comparison is made between fuzzy and non-fuzzy (Figure 4.50), it shows that fuzzy based approach with more fuzzy sets performs better for two evaluation metrics (Downtime rate (DR) and Energy Usage Efficiency (EUE). Fuzzy based power management system is capable to provide more energy efficient and less WSN sleeps due to low power storage. From these

results, the Energy Utilization Efficiency (EUE) is found to be 97%, which is better than the one in Section 4.5 with two fuzzy sets.

4.7.2 Fuzzy Vs Markov chain approaches

Average harvested energy, storage energy and packet loss probabilities are compared to those of Markov chain model presented in Chapter 3. The results shown in Figure 4.51 depicts that the average harvested energy based on the fuzzy logic was better (around 31% more energy was harvested) compared to the one of Markov based system with five states. This is because the Markov model-based energy harvesting system is prone to errors due to its probabilistic nature.

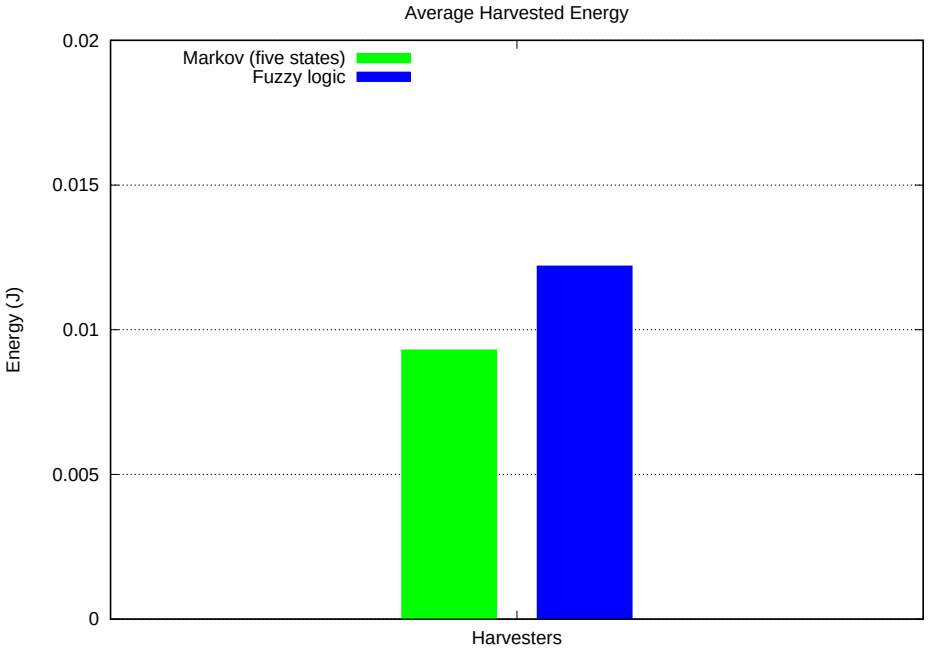

Figure 4.51. Average harvested energy

The results depicted in Figure 4.52 showed that the average stored energy based on the fuzzy logic increased by around 41% compared to the one stored in the Markov based system with five states. This is explained by the same reason that the Markov model-based energy harvesting systems are prone to errors due to its probabilistic approach. In terms of packet loss probabilities, the results depicted in Figure 4.53 demonstrated that the fuzzy logic had less chance of packets to be dropped due to energy availability.

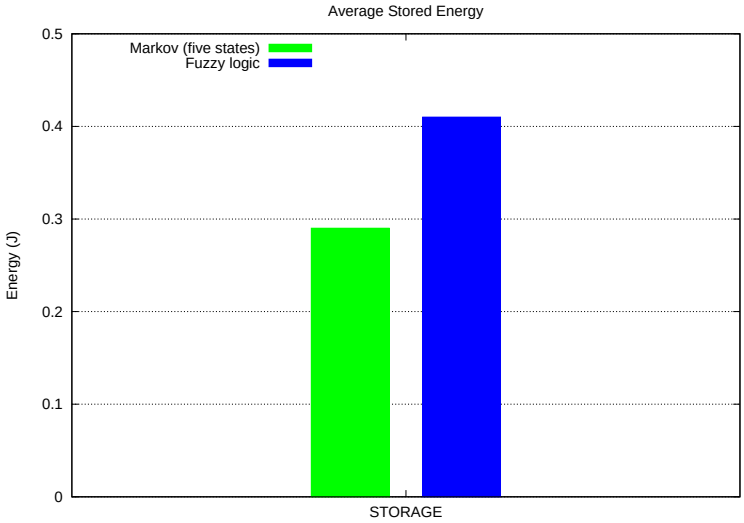

Figure 4.52. Average stored energy

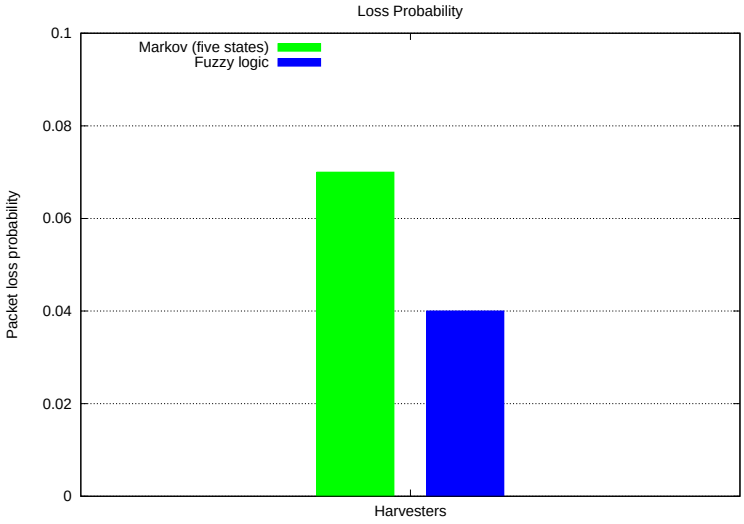

Figure 4.53. Packet loss probability

The packet loss probability was 0.04 compared to 0.07 in the Markov model based system. This is a decrease of around 43%. This is explained by more availability of energy in the battery in the Fuzzy system compared to that of the Markov model based energy harvesting systems.

4.8 Summary

This chapter was focused on energy harvesting based on LTSpice and NS3 integration. The LTspice and NS3 integration have been validated by deploying the Fuzzy logic control approach power management use case in energy harvesting. Results based on Fuzzy control logic expert system for power management system have indicated that the integration of LTspice and NS3 reach up to 97% of the energy efficiency utilization and can reduce the downtime ratio for up to 2%.

The energy efficiency utilization for the LTSpice and NS3 integration was found to be 97% compared to when fuzzy control was not implemented, which was around 59%. The downtime ratio was also small, at about 2% for the LTSpice and NS3 integration, this is better than that when fuzzy control was not implemented, which was around 41%.

More comparisons were made against the Markov model based system presented in Chapter 3. The results showed that the average harvested energy based on the fuzzy logic was better (around 31% more energy was harvested) compared to the one of Markov based system with five states. This is because the Markov model based energy harvesting system is prone to errors due to its probabilistic approach. The results showed that the average stored energy based on the fuzzy logic increased by around 41% compared to the one stored in the Markov based system with five states. This is explained by the same reason that the Markov model based energy harvesting systems are prone to errors due to its probabilistic approach. The results also showed that in terms of packet loss probabilities, the fuzzy logic had less chance of packets to be dropped due to sufficient energy availability. The packet loss probability decrease was around 43%. This is explained by more availability of energy in the battery in the Fuzzy system compared to that of the Markov model based energy harvesting systems.

5 LEACH BASED POWER MANAGEMENT

5.1 Overview

The previous two chapters demonstrated that harvesting power management based on the Markov chain model and fuzzy control logic was capable of efficiently harvesting power from piezoelectric and thermoelectric ambient sources. However, both systems used a direct communication mode. This thesis has demonstrated that energy can be further conserved by using cluster-based routing protocols such as LEACH to prolong the WSN nodes life during transmission.

All the proposed power management systems reviewed in the literature use conventional direct communication protocols and did not implement LEACH routing protocols. Each sensor in a direct communication protocol will transmit data directly to its corresponding base station. If the WSN nodes are far from the base station, then WSN nodes in direct communication will use more transmission power. As a result, the battery will quickly be drained and therefore, reduce the system lifetime. This thesis goes beyond these systems and implements LEACH protocol in both Markov chain model with five states and Fuzzy control logic with five sets in order to save more energy and extend WSN nodes life-time. The proposed systems are compared to direct communication protocols in both Markov chain model with five states and Fuzzy control logic with five sets to demonstrate its efficiency and suitability.

As proof of concept, LEACH based power management was implemented in both the integrated LTSpice and NS3 and Markovian Model based approaches. The Leach based routing protocol was simulated in NS3 with Piezoelectric as energy harvesting sources.

NS3 simulation results have demonstrated that the routing protocol based on LEACH increases energy efficiency in the aircraft's Piezoelectric energy harvesting process relative to a traditional direct communication protocol for both the five states Markov chain model and Fuzzy control expert system with five sets.

5.2 Simulation Setup

In this thesis, the simulation setup randomly deploys all WSN nodes in a given area, and the location of the base station is set up outside the deployment area of the WSN nodes. The initial simulation setup parameters also include deployment area. Note that the rest of other parameters remain the same because the LEACH routing protocol is used in both Markov chain model with five states and Fuzzy control logic with five sets as outlined in Chapter 3 and Chapter 4. The setup phase of the LEACH routing protocol is depicted in Figure 5.54 while the steady state phase is illustrated in Figure 5.55.

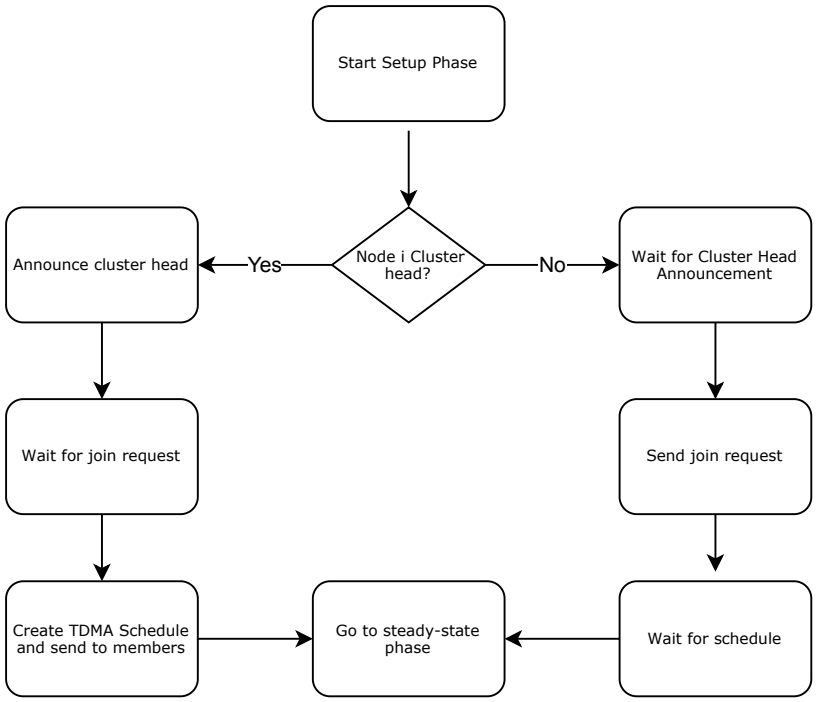

Figure 5.54. LEACH setup phase flowchart

- Routing protocol = LEACH
- Deployment area = 100x100m2
- Nodes deployment = random
- Total number of packets = 100000
- Packet size = 32 bytes
- Traffic type = Constant bit rate

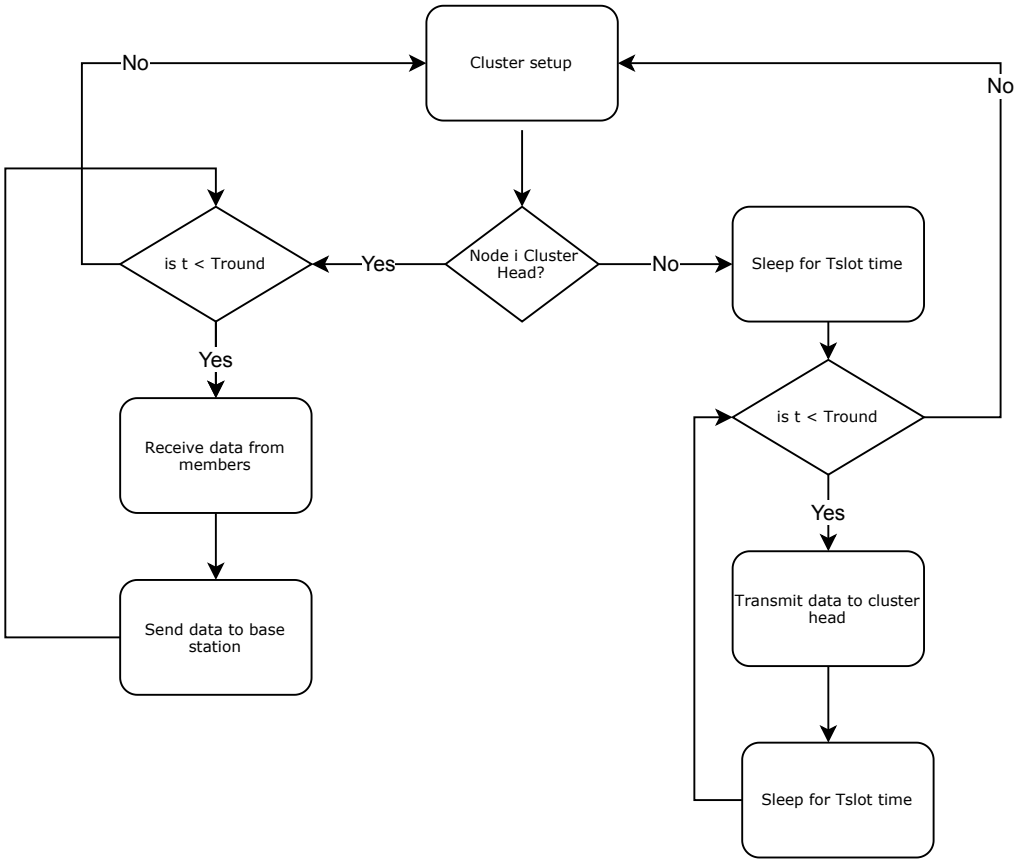

Figure 5.55. LEACH steady state phase flowchart

- 20 ms is chosen as the time slot
- The simulation is started at time 0 s
- The simulation duration is set to 600 s
- Threshold battery level for recharging = 0.11J
- Maximum battery level capacity = 0.15 J
- Wireless protocol used is 802.15.4 with TX = 0.0174J and RX = 0.0197J
- 0.11J is set as the threshold battery level for recharging the storage capacity
- 0.15 J is set as the threshold for maximum battery level capacity =
- The IEEE protocol for wireless communication is chosen as 802.15.4 with transmission and receiving power as TX = 0.0174J and RX = 0.0197J, respectively

- The piezoelectric generator randomly recharges the energy storage and in a uniformly distributed manner with the energy range of [0.0022 0.0044]. The energy is updated after every 1 ms. These values are based on the literature review presented in Section 2.1.1 [34, 35].
- Number of WSN nodes = 2000

5.3 Simulation Results

Figure 5.56 depicts the current harvested energy values during the 600 s duration of the simulation. The energy storage is recharged if the piezoelectric generator state is high. It can be observed that 0.00438774 J and 0.00014 J were the maximum and minimum harvested energy quantities reached during the simulation time at each time interval. It can be noted that if the threshold harvested energy of less than 0.0022 J is reached, then no recharging activities would take place.

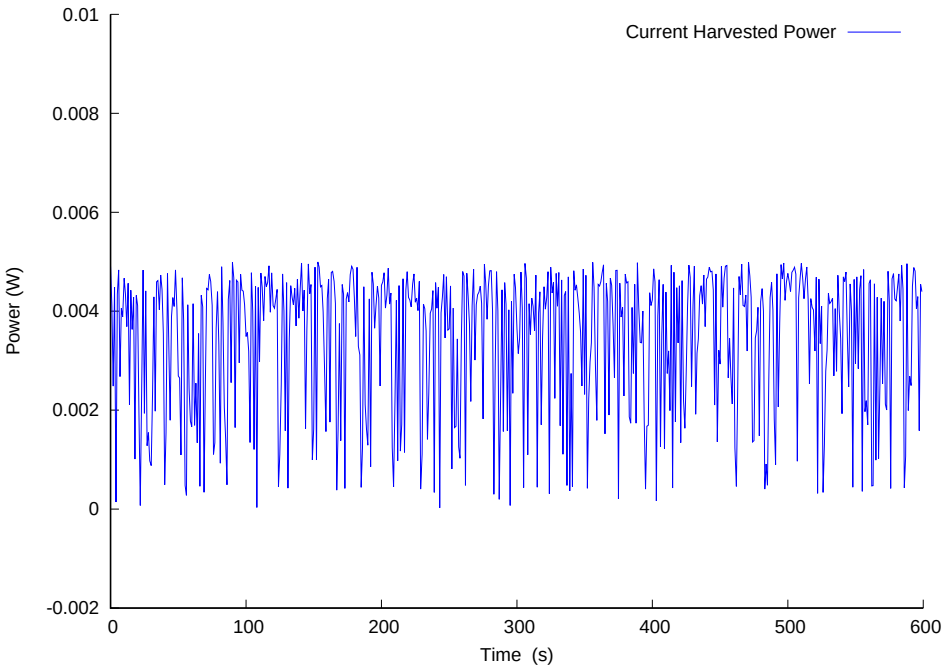

Figure 5.56. Current PEG harvested energy

Figure 5.57 depicts the current energy storage values during the 600 s duration of the simulation. The storage is recharged by using the piezoelectric generator if the minimum energy level is attained due to transmission activities of the transceiver. It can be observed that 0.136566 J and 0.107097

J were the maximum and minimum energy quantities reached during the simulation time at each time interval.

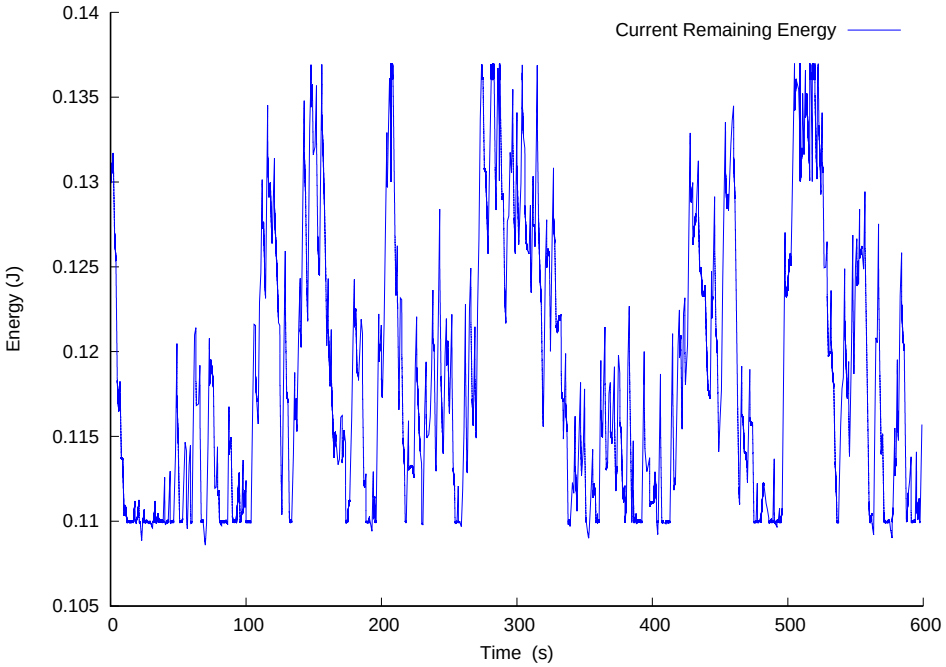

Figure 5.57. Current energy storage values

The LEACH routing protocol provides higher energy efficiency than direct communication. This is demonstrated in Figure 5.58. There are some scenarios where the WSN nodes deployment area is large, if this happens then some of the LEACH cluster head nodes will be far away from their base station, and hence, high transmission power will be used to get data from WSN nodes to the base station. This will cause these nodes to die quickly due to energy draining.

The LEACH routing protocol is capable to last for a higher number of rounds than the direct communication. Therefore, the lifetime of the WSN increases when using LEACH routing protocol for large coverage areas. Figure 5.58 depicts that the number of dead nodes in the WSN when using the LEACH protocol based on the Fuzzy control with five sets is less than the ones in direct communication based on the Markov model with five state. At 200 rounds, the number of dead nodes was 420 for LEACH protocol based on the fuzzy control with five sets, while the number of dead nodes for the direct communication based on the Markov model with five states was higher with 1100 (This was 62% improvement on Fuzzy control against Markov Model). In direct communication, nodes die sooner, and that leads to a low number

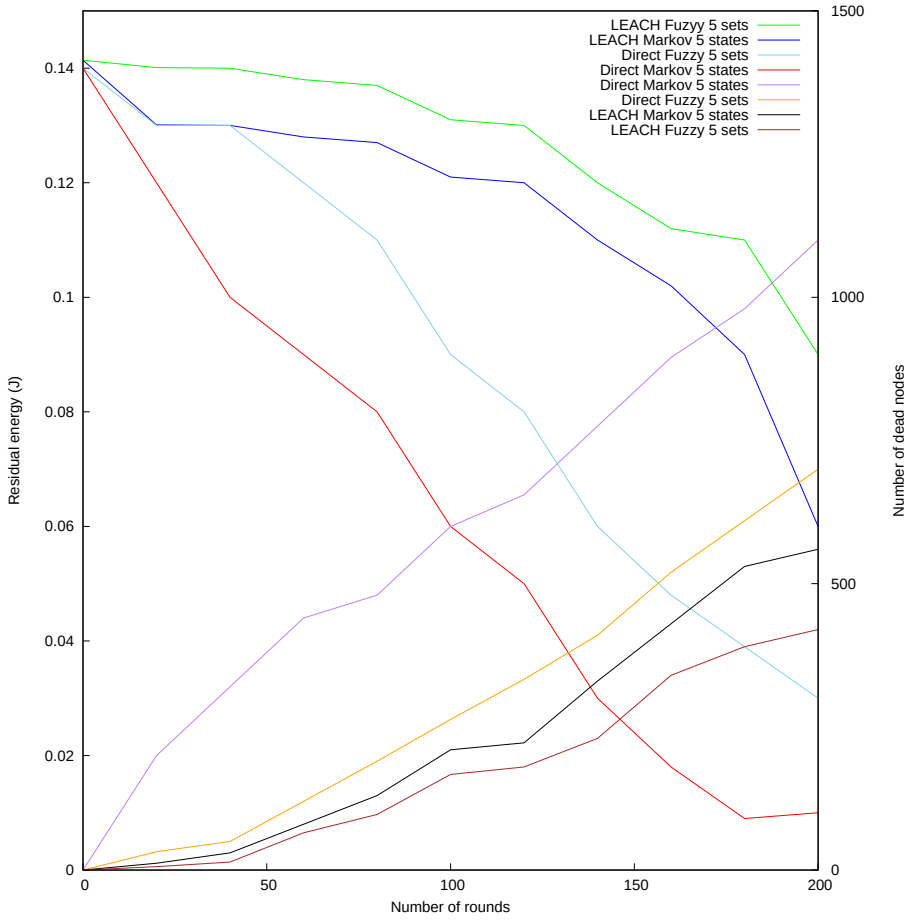

Figure 5.58. Total residual energy

of simulation rounds. For the residual energy, the residual energy dropped from 0.14J to around 0.09J for the LEACH protocol based on the Fuzzy control with five sets. This was better than the one with direct communication based on Markov model with 5 states where the residual energy dropped from 0.14J to around 0.01 in 200 rounds. The LEACH protocol based on the Fuzzy control with five sets had an improvement of about 90% over the direct communication based on the Markov model with five states.

Figure 5.59 depicts the total number of bits received by the base station over 200 rounds duration. The results show that a higher number of bits information is received by the base station over the LEACH routing protocol based on the Fuzzy control with five sets than the ones with direct communication based on the Markov model with five states. There were around 2,230,000 bits received when using the LEACH protocol based on the Fuzzy control compared to 1,600,000 bits when using LEACH protocol based on the Markov model with five states (This was 28% improvement on Fuzzy

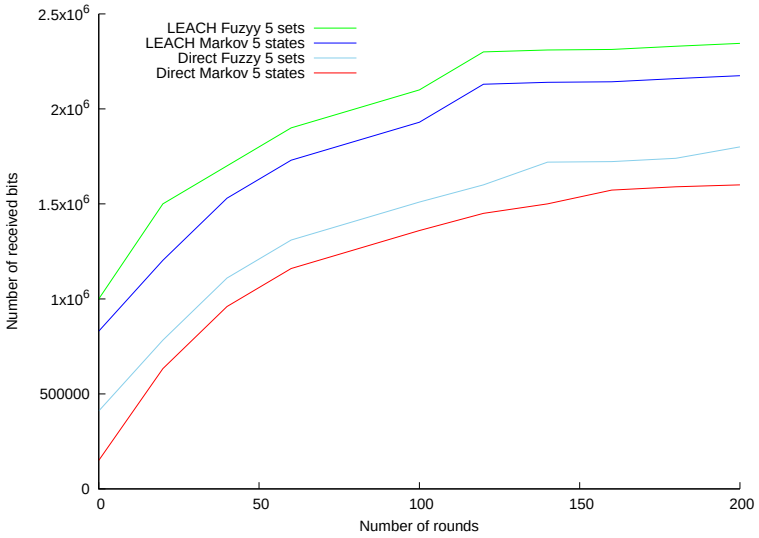

Figure 5.59. Number of rounds

control against Markov Model). The total number of nodes and clusters increases as the WSN deployment area is increased. Due to this, the energy consumption also increases because of the increased distances between nodes in direct communication. This eventually reduces the WSN lifetime. The lifetime of the WSN is relative longer under the LEACH routing protocol because of the less energy consumption.

Average harvested energy, storage energy and packet loss probabilities are compared to those of Fuzzy logic with five sets and Markov chain model with five states based systems who have used direct communication modes presented in Chapter 3 and 4.

The results shown in Figure 5.60 depict that the average harvested energy based on the LEACH routing protocol implemented with five Fuzzy sets was better (around 29% more energy was harvested) compared to the one of the Fuzzy logic with direct communication with five sets and around 46% better than that of the Markov based system with five states. This is because the LEACH based system was able to save more energy at each round. LEACH routing protocol implemented with five Markov states showed an increase of 17% compared to energy harvested in the Markov based system with five states.

The results depicted in Figure 5.61 showed that the average stored energy based on the LEACH routing protocol implemented with five Fuzzy sets increased by around 33% compared to the one stored in the Fuzzy logic system with five sets and around 53% to the Markov based system with

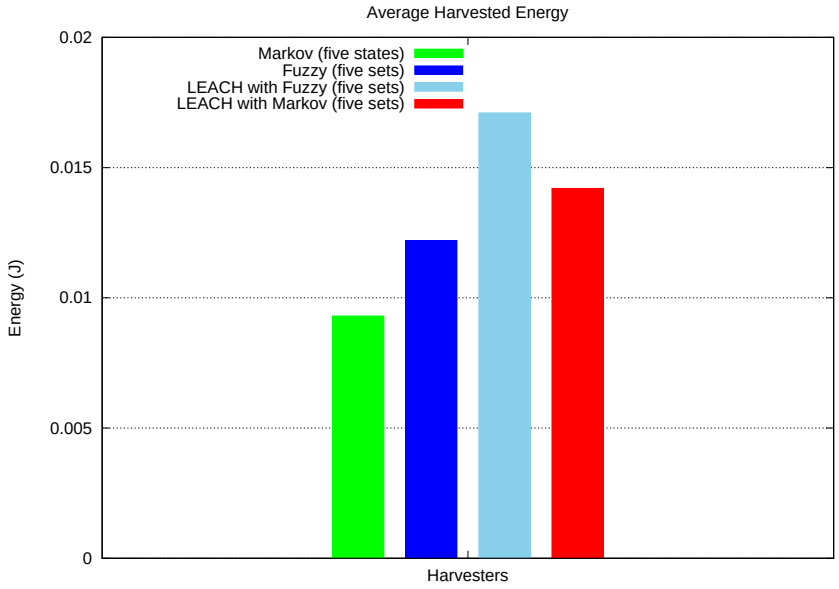

Figure 5.60. Average harvested energy

five states. LEACH routing protocol implemented with five Markov states showed an increase of 85% compared to stored energy in the Markov based system with five states.

This is explained by the reason that the LEACH routing protocol based energy harvesting system was able to save more energy at each round in the simulation.

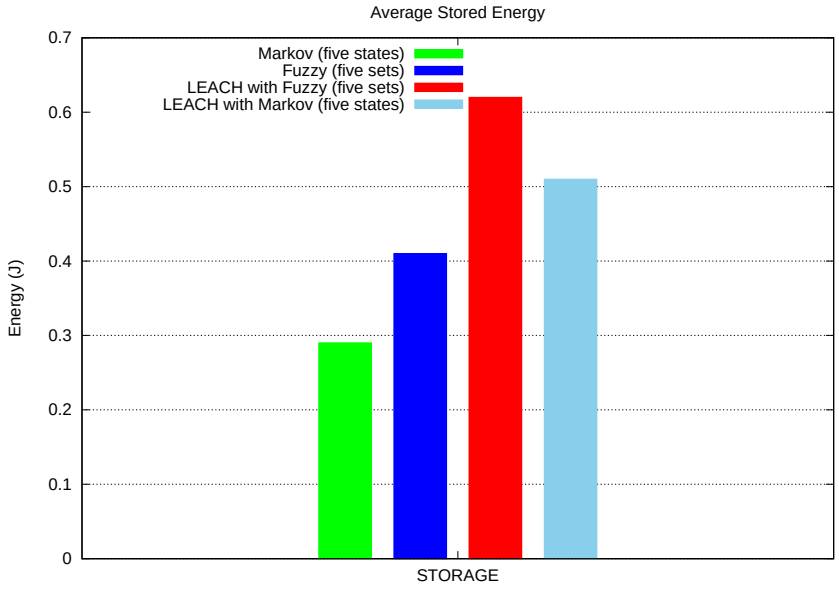

Figure 5.61. Average stored energy

In terms of packet loss probabilities, the results depicted in Figure 5.62

demonstrated that the LEACH routing based system implemented with five Fuzzy sets had less chance of packets to be dropped due to more energy availability compared to Fuzzy logic with five sets and Markov model with five states systems. The packet loss probability was 1% compared to 4% and 7% in the Fuzzy logic with five sets and Markov model based with five sets systems, respectively. LEACH routing protocol implemented with Markov with five states had 2% packet loss probability.

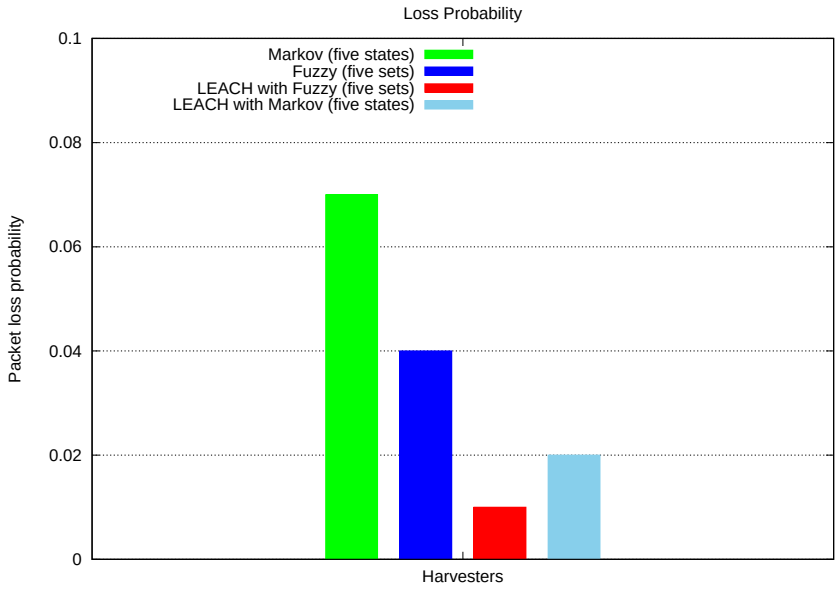

Figure 5.62. Packet loss probability

5.4 Summary

In this thesis, an NS3 simulation model for the LEACH routing protocol based on power management was proposed. The findings of the simulation results suggest that in contrast to direct communication mode between the Markov chain model with the five states and the Fuzzy logic control with five sets, the LEACH routing protocol demonstrated to be more energy efficient. The WSN nodes' lives were also improved. If the WSN deployment area is increased then the number WSN nodes and the number LEACH clusters increases, this increase will cause high transmission energy because of the increased distances between nodes in direct communication, and this eventually reduces the WSN nodes' lives. The lifetime of the WSN nodes is relative longer under the LEACH routing protocol because of less energy consumption.

It was observed that at the 200 rounds, the number of dead nodes was around 420 for LEACH protocol based on the Fuzzy control while for direct communication based on the Markov model, there were around 1100 dead nodes. In direct communication, nodes die sooner, and that leads to a low number of simulation rounds. For the residual energy, the residual energy dropped from 0.14J to around 0.09J for the LEACH protocol better than the one with direct communication based on the Markov chain model with five states where the residual energy dropped from 0.14J to around 0.01 in 200 rounds. The LEACH protocol based on the Fuzzy control has an improvement of about 62% over the direct communication based on the Markov chain model with five states.

Average harvested energy, storage energy and packet loss probabilities were also compared to those of Fuzzy logic with five sets and Markov chain model based with five states systems who have used direct communication modes presented in Chapter 3 and 4.

The results showed that the average harvested energy based on the LEACH routing protocol was better (around 29% more energy was harvested) compared to the one of the Fuzzy logic with direct communication with five sets and around 46% better than that of the Markov based system with five states. This is because the LEACH based system was able to save more energy at each round.

The results demonstrated that the average stored energy based on the LEACH routing protocol increased by around 33% compared to the one stored in the Fuzzy logic system with five sets and around 53% to the Markov based system with five states. This is explained by the same reason that the LEACH routing protocol based energy harvesting system was able to save more energy at each round in the simulation.

In terms of packet loss probabilities, the results also depicted that the LEACH routing based system had less chance of packets to be dropped due to more energy availability compared to Fuzzy logic and Markov model systems. The packet loss probability was 1% compared to 4% and 7% in the Fuzzy logic with five sets and Markov model based with five states systems, respectively.

These results have potential use in aircraft energy harvesting for structural health monitoring. Improved LEACH protocols were not tested in this thesis due to NS3 limitation, at the time of this thesis, NS3 only supported the standard LEACH protocol.

CONCLUSIONS and FUTURE WORK

6.1 Conclusions

The novel idea of the Integrated PEG and TEG approach was presented in this thesis. Two Markov states (Low and High) for energy harvesting, storage and data queue were initially used. The two Markov states did not capture the real world scenario because there were some states between Low and High. Five Markov states (Low, Low Medium, Medium, High Medium and High) were deployed to closely represent the real world scenario instead of two Low and High states.

More average energy storage was obtained in the integrated approach with five Markov states compared to only two states transitions. There was an increase of about 64% in energy storage with five Markov states compared to only two Markov states.

The packet loss probability of the integrated PEG and TEG approach with five Markov states was found to be 2% compared to 23% for two Markov states. These results demonstrated that the integrated PEG and TEG approach together with effective stochastic scheduling with five Markov states obtained higher harvested energy. Higher harvested energy meant high energy storage and less frequent transceiver sleep modes.

The results also showed that the integrated approach with five Markov states harvested more energy than the rest. It was 64% more harvested energy than the one with two Markov states.

The implemented TEG and PEG integrated approach has achieved the following objectives set out in Chapter 1.

- Conduct literature review on state of the art in vibration and thermal energy harvesters for WSN systems in aircraft AHM system and energy-efficient communication and routing protocols that can be deployed to minimize power consumption in aircraft for AHM systems.
- Formulate a Markov model and fuzzy expert system for node-level decisions for battery health, recharging urgency and priority, and data transmission.

The novel energy harvesting based on LTSpice and NS3 integration was also proposed in this thesis. The LTspice and NS3 integration have been validated by deploying the Fuzzy logic control approach power management use case in energy harvesting. Results based on Fuzzy control logic expert system for power management system have indicated that the integration of LTspice and NS3 reached up to 97% of the energy efficiency utilization and could reduce the downtime ratio for up to 2%.

The energy efficiency utilization for the LTSpice and NS3 integration was found to be 97% compared to when fuzzy control was not implemented, which was around 59%. The downtime ratio was also very small, at about 2% for the LTSpice and NS3 integration, this is better than the one without fuzzy control, which was around 41%.

Further comparisons were made against the Markov model based system presented in Chapter 3. The results showed that the average harvested energy based on the fuzzy logic was better (around 31% more energy was harvested) than the one of Markov based system with five states. This was because the Markov model based energy harvesting system was prone to errors due to its probabilistic approach. The results showed that the average stored energy based on the fuzzy logic increased by around 41% compared to the one stored in the Markov based system with five states. This is because the Markov model based energy harvesting systems was prone to errors due to their probabilistic approach. The results also showed that in terms of packet loss probabilities, the fuzzy logic had less chance of packets to be dropped due to sufficient energy availability. The packet loss probability decrease was around 43%. This is because there is more available of energy in the battery in the Fuzzy system than that of the Markov model based energy harvesting systems.

The implemented NS3 and LTspice integration based on Fuzzy control has achieved the following objectives set out in Chapter 1.

- Conduct literature review on state of the art in vibration and thermal energy harvesters for WSN systems in aircraft AHM system and energy-efficient communication and routing protocols that can be deployed to minimize power consumption in aircraft for AHM systems.
- Develop and simulate Markov and fuzzy expert systems by using LT-

spice and NS3 simulation tools.

- Conduct comparative analysis between Markov prediction models and fuzzy expert systems for power management systems in aircraft for AHM.

In this thesis, an NS3 simulation model for the LEACH routing protocol based on power management was proposed. The findings of the simulation results suggest that in contrast to direct communication mode between the Markov chain model with the five states and the Fuzzy logic control with five sets, the LEACH routing protocol demonstrated to be more energy efficient. The WSN nodes' lifetime was also improved.

If the WSN deployment area is extended then the energy consumption will increase because of the increased distance between wireless nodes. This scenario affects direct communication more than LEACH routing protocol. Therefore, the lifetime of the WSN nodes is relative longer under the LEACH routing protocol than in direct communication.

It was observed that in 200 rounds, the number of dead nodes was around 700 for LEACH protocol while for direct communication, there were around 1100 dead nodes (almost 40% more dead WSN nodes than in the LEACH routing protocol). For direct communication, nodes die sooner, and that leads to a low number of simulation rounds. The residual energy dropped from 0.14J to around 0.09J for the LEACH protocol, this was better than the one with direct communication in both Markov chain model with five states and Fuzzy control with five sets. For the Fuzzy control logic with five sets, the residual energy dropped from 0.14J to around 0.01 in 200 rounds. In terms of residual energy, the LEACH protocol has an improvement of about 90% over the direct communication model in both Markov chain model with five states and Fuzzy control logic with five sets.

Average harvested energy, storage energy and packet loss probabilities were also compared to those of Fuzzy logic with five sets and Markov chain model based with five states systems who have used direct communication modes presented in Chapter 3 and 4.

The results showed that the average harvested energy based on the LEACH routing protocol was better (around 29% more energy was harvested) com-

pared to the one of the Fuzzy logic with direct communication with five sets and around 46% better than that of the Markov based system with five states. This is because the LEACH based system was able to save more energy at each round.

The results demonstrated that the average stored energy based on the LEACH routing protocol implemented with five Fuzzy sets increased by around 33% compared to the one stored in the Fuzzy logic system with five sets and around 53% to the Markov based system with five states. This is because the LEACH routing protocol based energy harvesting system was able to save more energy at each round in the simulation.

In terms of packet loss probabilities, the results also depicted that the LEACH routing based system implemented with five Fuzzy sets had less chance of packets to be dropped due to more energy availability compared to Fuzzy logic and Markov model systems. The packet loss probability was 1% compared to 4% and 7% in the Fuzzy logic with five sets and Markov model based on five states systems, respectively. The implemented LEACH routing protocol with five Fuzzy sets has achieved the following objectives set out in Chapter 1.

- Conduct literature review on state of the art in vibration and thermal energy harvesters for WSN systems in aircraft AHM system and energy-efficient communication and routing protocols that can be deployed to minimize power consumption in aircraft for AHM systems.
- Develop and simulate energy-efficient communication and routing protocols for AHM.
- Conduct comparative analysis between Markov prediction models and fuzzy expert systems for power management systems in aircraft for AHM.

These results have potential use in aircraft energy harvesting for structural health monitoring. Improved LEACH protocols were not tested in this thesis due to NS3 limitation, at the time of this thesis, NS3 only supported the standard LEACH protocol.

6.2 Strengths and Limitations

As with several other studies, this thesis has its strengths and limitations that may have influenced the results.

The strength of this thesis is that it has integrated two popular simulators NS3 and LTSpice and propose an expert based power management system in WSN-based aircraft management system. Results based on the fuzzy control logic expert system for power management system have indicated that the integration of LTspice and NS3 increased the energy efficiency utilization and reduced the downtime ratio.

Another strength is the novel Markovian chain based integration of PEG and TEG. The results have shown that the integrated approach with five Markov states harvested more energy than the Markov chain model with two states.

The deployment of LEACH routing protocol in AHM systems were another strength in this thesis. The deployment of LEACH routing protocol has proved to efficient in conserving more energy and hence, prolong the lifetime of the wireless sensor nodes.

The proposed system could have been implemented on hardware in order to validate the simulation results and the real implementation results. This is a major limitation of this thesis. However, this could be considered as future work.

6.3 Recommendations for Future Work

This thesis has accomplished the original objectives that were set in Chapter 1. This thesis has demonstrated the theoretical and practical effectiveness of using expert based systems for power management in which the energy is harvested from vibration and therm ambient sources.

Although all objectives set out in Chapter 1 were achieved, several issues require further work,

- Implementation of the proposed expert systems power management in real hardware. This will be able to efficiently validate the simulation results against the real world experimental results.

- Other low power communication protocols could be used in the future, such as LORA and 6LoWPAN. These protocols could be used to further reduce energy consumption in WSNs.

References

[1] R. Vullers, R. van Schaijk, I. Doms, C. Van Hoof, and R. Mertens, "Micropower energy harvesting," *Solid-State Electronics*, vol. 53, no. 7, pp. 684–693, 2009.

[2] H. Wu, S. Nabar, and R. Poovendran, "An energy framework for the network simulator 3 (ns-3)," in *Proceedings of the 4th international ICST conference on simulation tools and techniques*, pp. 222–230, 2011.

[3] L. Lamont, M. Toulgoat, M. Deziel, and G. Patterson, "Tiered wireless sensor network architecture for military surveillance applications," in *The Fifth International Conference on Sensor Technologies and Applications, SENSORCOMM*, pp. 288–294, 2011.

[4] V. Hejlova and V. Vozenilek, "Wireless sensor network components for air pollution in urban environment: Criteria and analysis for their selection," *Wireless Sensor Networks*, pp. 229–240.

[5] M. Keshtgari and A. Deljoo, "A wireless sensor network solution for precision agriculture based on zigbee technology," 2011.

[6] A. Pascale, M. Nicoli, F. Deflorio, B. Dalla Chiara, and U. Spagnolini, "Wireless sensor networks for traffic management and road safety," *IET Intelligent Transport Systems*, vol. 6, no. 1, pp. 67–77, 2012.

[7] J. Guevara, F. Barrero, E. Vargas, J. Becerra, and S. Toral, "Environmental wireless sensor network for road traffic applications," *IET Intelligent Transport Systems*, vol. 6, no. 2, pp. 177–186, 2012.

[8] M. A. Kumbhar, "Wireless sensor networks: A solution for smart transportation," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 4, 2012.

[9] J. I. Huircán, C. Muñoz, H. Young, L. Von Dossow, J. Bustos, G. Vivallo, and M. Toneatti, "Zigbee-based wireless sensor network localization for cattle monitoring in grazing fields," *Computers and Electronics in Agriculture*, vol. 74, no. 2, pp. 258–264, 2010.

[10] G. Sasikumar, H. V. Ramamoorthy, and S. N. Mohamed, "An analysis on animal tracking system using wireless sensors," *International*

Journal of Advanced Research in computer science and software engineering, vol. 4, no. 9, pp. 155–162, 2014.

[11] A. Gaddam, S. C. Mukhopadhyay, and G. S. Gupta, "Elder care based on cognitive sensor network," *IEEE sensors journal*, vol. 11, no. 3, pp. 574–581, 2010.

[12] M. Aldeer, R. P. Martin, and R. E. Howard, "Pillsense: Designing a medication adherence monitoring system using pill bottle-mounted wireless sensors," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.

[13] S. Jena and A. Gupta, "Embedded sensors for health monitoring of an aircraft," in *Sensors for Automotive and Aerospace Applications*, pp. 77–91, Springer, 2019.

[14] X. Zhao, H. Gao, G. Zhang, B. Ayhan, F. Yan, C. Kwan, and J. L. Rose, "Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. defect detection, localization and growth monitoring," *Smart materials and structures*, vol. 16, no. 4, p. 1208, 2007.

[15] X. Qing, W. Li, Y. Wang, and H. Sun, "Piezoelectric transducer-based structural health monitoring for aircraft applications," *Sensors*, vol. 19, no. 3, p. 545, 2019.

[16] P. Sundriyal and S. Bhattacharya, "Energy harvesting techniques for powering wireless sensor networks in aircraft applications: A review," in *Sensors for Automotive and Aerospace Applications*, pp. 55–76, Springer, 2019.

[17] G. Baxter, P. Srisaeng, and G. Wild, "The role of the airbus a380-800 aircraft in a full-service network airline's commercial operations and route network design: the case of thai airways international," *Aeron Aero Open Access J*, vol. 2, no. 4, pp. 223–236, 2018.

[18] I. Dörfler and O. Baumann, "Learning from a drastic failure: the case of the airbus a380 program," *Industry and Innovation*, vol. 21, no. 3, pp. 197–214, 2014.

[19] B. Marr, "That's data science: Airbus puts 10,000 sensors in every single wing," *URL: http://www. datasciencecentral.*

com/profiles/blogs/that-s-data-science-airbus-puts-10-000-sensors-in-every-single/(visited on 2017-03-14), 2015.

[20] A. T. A. Group *et al.*, "Facts and figures," 2014.

[21] M. Al Nuaimi, F. Sallabi, and K. Shuaib, "A survey of wireless multimedia sensor networks challenges and solutions," in *Innovations in Information Technology (IIT), 2011 International Conference on*, pp. 191–196, IEEE, 2011.

[22] M. Q. Le, J.-F. Capsal, M. Lallart, Y. Hebrard, A. Van Der Ham, N. Reffe, L. Geynet, and P.-J. Cottinet, "Review on energy harvesting for structural health monitoring in aeronautical applications," *Progress in Aerospace Sciences*, vol. 79, pp. 147–157, 2015.

[23] Y. Lu, A. Savvaris, A. Tsourdos, and M. Bevilacqua, "Vibration energy harvesters for wireless sensor networks for aircraft health monitoring," in *Metrology for Aerospace (MetroAeroSpace), 2016 IEEE*, pp. 25–32, IEEE, 2016.

[24] T.-B. Xu, "Energy harvesting using piezoelectric materials in aerospace structures," in *Structural Health Monitoring (SHM) in Aerospace Structures*, pp. 175–212, Elsevier, 2016.

[25] K. A. Borup, J. De Boor, H. Wang, F. Drymiotis, F. Gascoin, X. Shi, L. Chen, M. I. Fedorov, E. Müller, B. B. Iversen, *et al.*, "Measuring thermoelectric transport properties of materials," *Energy & Environmental Science*, vol. 8, no. 2, pp. 423–435, 2015.

[26] Y. Wu, L. Zuo, J. Chen, and J. A. Klein, "A model to analyze the device level performance of thermoelectric generator," *Energy*, vol. 115, pp. 591–603, 2016.

[27] A. Elefsiniotis, N. Kokorakis, T. Becker, and U. Schmid, "A thermoelectric-based energy harvesting module with extended operational temperature range for powering autonomous wireless sensor nodes in aircraft," *Sensors and Actuators A: Physical*, vol. 206, pp. 159–164, 2014.

[28] C. Vankecke, L. Assouère, A. Wang, P. Durand-Estèbe, F. Caignet, J.-M. Dilhac, and M. Bafleur, "Multisource and battery-free energy har-

vesting architecture for aeronautics applications," *IEEE Transactions on Power Electronics*, vol. 30, no. 6, pp. 3215–3227, 2015.

[29] P. Panda, "Piezoceramic materials and devices for aerospace applications," in *Aerospace Materials and Material Technologies*, pp. 501–518, Springer, 2017.

[30] L. Allmen, G. Bailleul, T. Becker, J.-D. Decotignie, M. Kiziroglou, C. Leroux, P. Mitcheson, J. Müller, D. Piguet, T. Toh, *et al.*, "Aircraft strain wsn powered by heat storage harvesting," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 9, pp. 7284–7292, 2017.

[31] C. S. Clemente, D. Davino, G. Maddaloni, M. R. Pecce, and C. Visone, "A magnetostrictive energy harvesting system for bridge structural health monitoring," in *Advances in Science and Technology*, vol. 101, pp. 20–25, Trans Tech Publ, 2017.

[32] G. Zhou, L. Huang, W. Li, and Z. Zhu, "Harvesting ambient environmental energy for wireless sensor networks: a survey," *Journal of Sensors*, vol. 2014, 2014.

[33] J.-H. Lee, J. Kim, T. Y. Kim, M. S. Al Hossain, S.-W. Kim, and J. H. Kim, "All-in-one energy harvesting and storage devices," *Journal of Materials Chemistry A*, vol. 4, no. 21, pp. 7983–7999, 2016.

[34] R. D. Schaller and V. I. Klimov, "High efficiency carrier multiplication in pbse nanocrystals: implications for solar energy conversion," *Physical review letters*, vol. 92, no. 18, p. 186601, 2004.

[35] N. Elvin and A. Erturk, *Advances in energy harvesting methods*. Springer Science & Business Media, 2013.

[36] H. S. Kim, J.-H. Kim, and J. Kim, "A review of piezoelectric energy harvesting based on vibration," *International journal of precision engineering and manufacturing*, vol. 12, no. 6, pp. 1129–1141, 2011.

[37] M. Mouis, E. Chávez-Ángel, C. Sotomayor-Torres, F. Alzina, M. V. Costache, A. G. Nassiopoulou, K. Valalaki, E. Hourdakis, S. O. Valenzuela, B. Viala, *et al.*, "Thermal energy harvesting," *Beyond-CMOS Nanodevices 1*, pp. 135–219, 2014.

[38] D. Carli, D. Brunelli, D. Bertozzi, and L. Benini, “A high-efficiency wind-flow energy harvester using micro turbine,” in *Power electronics electrical drives automation and motion (SPEEDAM), 2010 international symposium on*, pp. 778–783, IEEE, 2010.

[39] S. Li, J. Yuan, and H. Lipson, “Ambient wind energy harvesting using cross-flow fluttering,” 2011.

[40] J. Yang, J. Chen, Y. Liu, W. Yang, Y. Su, and Z. L. Wang, “Triboelectrification-based organic film nanogenerator for acoustic energy harvesting and self-powered active acoustic sensing,” *ACS nano*, vol. 8, no. 3, pp. 2649–2657, 2014.

[41] F. Liu, A. Phipps, S. Horowitz, K. Ngo, L. Cattafesta, T. Nishida, and M. Sheplak, “Acoustic energy harvesting using an electromechanical helmholtz resonator,” *The Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 1983–1990, 2008.

[42] A. Khaligh, P. Zeng, and C. Zheng, “Kinetic energy harvesting using piezoelectric and electromagnetic technologies—state of the art,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 3, pp. 850–860, 2010.

[43] H. Reinisch, S. Gruber, H. Unterassinger, M. Wiessflecker, G. Hofer, W. Pribyl, and G. Holweg, “An electro-magnetic energy harvesting system with 190 nw idle mode power consumption for a baw based wireless sensor node,” *IEEE Journal of solid-state circuits*, vol. 46, no. 7, pp. 1728–1741, 2011.

[44] S. P. Beeby, M. J. Tudor, and N. White, “Energy harvesting vibration sources for microsystems applications,” *Measurement science and technology*, vol. 17, no. 12, p. R175, 2006.

[45] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green, “Energy harvesting from human and machine motion for wireless electronic devices,” *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1457–1486, 2008.

[46] S. P. Beeby, R. Torah, M. Tudor, P. Glynne-Jones, T. O’donnell, C. Saha, and S. Roy, “A micro electromagnetic generator for vibration

energy harvesting," *Journal of Micromechanics and microengineering*, vol. 17, no. 7, p. 1257, 2007.

[47] G. Gautschi, "Background of piezoelectric sensors," in *Piezoelectric Sensorics*, pp. 5–11, Springer, 2002.

[48] R. E. Pelrine, R. D. Kornbluh, and J. P. Joseph, "Electrostriction of polymer dielectrics with compliant electrodes as a means of actuation," *Sensors and Actuators A: Physical*, vol. 64, no. 1, pp. 77–85, 1998.

[49] H. Nagata, M. Yoshida, Y. Makiuchi, and T. Takenaka, "Large piezoelectric constant and high curie temperature of lead-free piezoelectric ceramic ternary system based on bismuth sodium titanate-bismuth potassium titanate-barium titanate near the morphotropic phase boundary," *Japanese journal of applied physics*, vol. 42, no. 12R, p. 7401, 2003.

[50] T. Soyata, L. Copeland, and W. Heinzelman, "Rf energy harvesting for embedded systems: A survey of tradeoffs and methodology," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 22–57, 2016.

[51] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *Southeastcon, 2008. IEEE*, pp. 442–447, IEEE, 2008.

[52] M.-G. Kang, W.-S. Jung, C.-Y. Kang, and S.-J. Yoon, "Recent progress on pzt based piezoelectric energy harvesting technologies," in *Actuators*, vol. 5, p. 5, Multidisciplinary Digital Publishing Institute, 2016.

[53] Y. Luo, R. Gan, S. Wan, R. Xu, and H. Zhou, "Design and analysis of a mems-based bifurcate-shape piezoelectric energy harvester," *Aip Advances*, vol. 6, no. 4, p. 045319, 2016.

[54] G. Zhang, S. Gao, and H. Liu, "A utility piezoelectric energy harvester with low frequency and high-output voltage: Theoretical model, experimental verification and energy storage," *AIP Advances*, vol. 6, no. 9, p. 095208, 2016.

[55] M. Guan, K. Wang, D. Xu, and W.-H. Liao, "Design and experimental investigation of a low-voltage thermoelectric energy harvesting sys-

tem for wireless sensor nodes," *Energy conversion and management*, vol. 138, pp. 30–37, 2017.

[56] C. Botteron, D. Briand, B. Mishra, G. Tasselli, P. Janphuang, F.-J. Haug, A. Skrivervik, R. Lockhart, C. Robert, N. F. de Rooij, *et al.*, "A low-cost uwb sensor node powered by a piezoelectric harvester or solar cells," *Sensors and Actuators A: Physical*, vol. 239, pp. 127–136, 2016.

[57] A. Nechibvute, A. Chawanda, and P. Luhanga, "Experimental study of self-powered electronic circuits for piezoelectric energy harvesting: paper presented at a conference held on 14-16 july 2015 at elephant hills resort, victoria falls, zimbabwe," 2016.

[58] A. Minnich, M. Dresselhaus, Z. Ren, and G. Chen, "Bulk nanostructured thermoelectric materials: current research and future prospects," *Energy & Environmental Science*, vol. 2, no. 5, pp. 466–479, 2009.

[59] J. Tervo, A. Manninen, R. Ilola, and H. Hänninen, "State-of-the-art of thermoelectric materials processing," *VTT Technical Research Center of Finland, Vuorimiehentie*, 2009.

[60] E. Schwyter, W. Glatz, L. Durrer, and C. Hierold, "Flexible micro thermoelectric generator based on electroplated bi 2+ x te 3- x," in *Design, Test, Integration and Packaging of MEMS/MOEMS, 2008. MEMS/-MOEMS 2008. Symposium on*, pp. 46–48, IEEE, 2008.

[61] H. Bottner, "Thermoelectric micro devices: current state, recent developments and future aspects for technological progress and applications," in *Thermoelectrics, 2002. Proceedings ICT'02. Twenty-First International Conference on*, pp. 511–518, IEEE, 2002.

[62] M. Kishi, H. Nemoto, T. Hamao, M. Yamamoto, S. Sudou, M. Mandai, and S. Yamamoto, "Micro thermoelectric modules and their application to wristwatches as an energy source," in *Thermoelectrics, 1999. Eighteenth International Conference on*, pp. 301–307, IEEE, 1999.

[63] M. Kocoloski, C. Eger, R. Mccarty, K. P. Hallinan, and J. K. Kissock, "Industrial solid-state energy harvesting: Mechanisms and examples," 2007.

[64] J. Eakburanawat and I. Boonyaroonate, “Development of a thermoelectric battery-charger with microcontroller-based maximum power point tracking technique,” *Applied Energy*, vol. 83, no. 7, pp. 687–704, 2006.

[65] D. Mastbergen, B. Willson, and S. Joshi, “Producing light from stoves using a thermoelectric generator,” *Ethos*, vol. 2005, 2005.

[66] H. A. Sodano, G. E. Simmers, R. Dereux, and D. J. Inman, “Recharging batteries using energy harvested from thermal gradients,” *Journal of Intelligent material systems and structures*, vol. 18, no. 1, pp. 3–10, 2007.

[67] L. Mateu, C. Codrea, N. Lucas, M. Pollak, and P. Spies, “Energy harvesting for wireless communication systems using thermogenerators,” in *Proc. of the XXI Conference on Design of Circuits and Integrated Systems (DCIS), Barcelona, Spain*, pp. 22–24, 2006.

[68] J. M. Damaschke, “Design of a low-input-voltage converter for thermoelectric generator,” *IEEE Transactions on Industry Applications*, vol. 33, no. 5, pp. 1203–1207, 1997.

[69] X. Lu and S.-H. Yang, “Solar energy harvesting for zigbee electronics,” in *Sustainability in Energy and Buildings*, pp. 19–27, Springer, 2009.

[70] G. Biccario, M. De Vittorio, and S. D’Amico, “Fluids energy harvesting system with low cut-in velocity piezoelectric mems,” in *2017 IEEE International Conference on IC Design and Technology (ICICDT)*, pp. 1–4, IEEE, 2017.

[71] A. Yamawaki and S. Serikawa, “Battery life estimation of sensor node with zero standby power consumption,” in *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pp. 166–172, IEEE, 2016.

[72] H. A. Sodano, D. J. Inman, and G. Park, “Comparison of piezoelectric energy harvesting devices for recharging batteries,” *Journal of intelligent material systems and structures*, vol. 16, no. 10, pp. 799–807, 2005.

[73] P. Kinney *et al.*, "Zigbee technology: Wireless control that simply works," in *Communications design conference*, vol. 2, pp. 1–7, 2003.

[74] D. Gislason, *ZigBee wireless networking*. Newnes, 2008.

[75] P. McDermott-Wells, "What is bluetooth?," *IEEE potentials*, vol. 23, no. 5, pp. 33–35, 2004.

[76] J. Kardach, "Bluetooth architecture overview," *Intel Technology Journal*, vol. 2, p. 7, 2000.

[77] A. Bensky, *Wireless positioning technologies and applications*. Artech House, 2016.

[78] J. Haartsen, "Bluetooth-the universal radio interface for ad hoc, wireless connectivity," *Ericsson review*, vol. 3, no. 1, pp. 110–117, 1998.

[79] P. B. Tan, M. J. Baptist, K. H. Wong, H. Lui, and X. Li, "Bluetooth pairing system, method, and apparatus," Mar. 28 2017. US Patent 9,609,677.

[80] G. A. Naidu and J. Kumar, "Wireless protocols: Wi-fi son, bluetooth, zigbee, z-wave, and wi-fi," in *Innovations in Electronics and Communication Engineering*, pp. 229–239, Springer, 2019.

[81] I. Khemapech, I. Duncan, and A. Miller, "A survey of wireless sensor networks technology," in *Proc. of The 6th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting*, vol. 32, 2005.

[82] N. Hariharan and S. Sreelekshmi, "A survey on wireless sensor networks," *International Journal of Scientific Engineering and Science*, vol. 1, no. 11, 2017.

[83] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in iot operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, 2018.

[84] S. P. Singh and S. Sharma, "A survey on cluster based routing protocols in wireless sensor networks," *Procedia computer science*, vol. 45, pp. 687–695, 2015.

[85] M. Hammoudeh and R. Newman, “Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance,” *Information Fusion*, vol. 22, pp. 3–15, 2015.

[86] S. S. Iyengar and R. R. Brooks, *Distributed Sensor Networks: Sensor Networking and Applications (Volume Two)*. CRC press, 2016.

[87] Y. Yao, Q. Cao, and A. V. Vasilakos, “Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 3, pp. 810–823, 2015.

[88] J. Angel and T. Ramesh, “Review on different approaches on trust establishment in wireless sensor networks,” 2017.

[89] Q. Xian and Y. Long, “An enhanced greedy perimeter stateless routing algorithm for wireless sensor network,” in *2016 IEEE international conference of online analysis and computing science (ICOACS)*, pp. 181–184, IEEE, 2016.

[90] A. R. Prusty, S. Sethi, and A. K. Nayak, “A hybrid multi-hop mobility assisted heterogeneous energy efficient cluster routing protocol for wireless ad hoc sensor networks,” *Journal of High Speed Networks*, vol. 22, no. 4, pp. 265–280, 2016.

[91] L. Rosyidi and R. F. Sari, “Energy harvesting aware protocol for 802.11-based internet of things network,” in *2016 IEEE Region 10 Conference (TENCON)*, pp. 1325–1328, IEEE, 2016.

[92] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pp. 10–pp, IEEE, 2000.

[93] V. Bahl and A. Bhola, “Enhanced energy efficient cluster head selection algorithm for improvement in network lifetime for wireless sensor network,” *International Journal of Information Systems & Management Science, Forthcoming*, 2018.

[94] F. Xiangning and S. Yulin, “Improvement on leach protocol of wireless sensor network,” in *Sensor Technologies and Applications, 2007.*

SensorComm 2007. International Conference on, pp. 260–264, IEEE, 2007.

[95] H. Junping, J. Yuhui, and D. Liang, "A time-based cluster-head selection algorithm for leach," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pp. 1172–1176, IEEE, 2008.

[96] M. C. M. Thein and T. Thein, "An energy efficient cluster-head selection for wireless sensor networks," in *Intelligent systems, modelling and simulation (ISMS), 2010 international conference on*, pp. 287–291, IEEE, 2010.

[97] F. Zhang and Q. Wang, "A probability optimize clustering routing algorithm for wsns," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 4, pp. 348–352, IEEE, 2011.

[98] A. K. Paul and T. Sato, "Effective data gathering and energy efficient communication protocol in wireless sensor network," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pp. 1–5, IEEE, 2011.

[99] N. A. Pantazis, "Survey on power control issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 4, pp. 86–107, 2007.

[100] R. Pell, "Free spice software exploits multicore processors," *EE Times. Retrieved*, 2011.

[101] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.

[102] D. P. Van, B. P. Rimal, J. Chen, P. Monti, L. Wosinska, and M. Maier, "Power-saving methods for internet of things over converged fiber-wireless access networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 166–175, 2016.

[103] S. Roundy, E. S. Leland, J. Baker, E. Carleton, E. Reilly, E. Lai, B. Otis, J. M. Rabaey, P. K. Wright, and V. Sundararajan, "Improving power output for vibration-based energy scavengers," *IEEE Pervasive Computing*, vol. 4, pp. 28–36, Jan 2005.

[104] K. J. Kim, F. Cottone, S. Goyal, and J. Punch, “Energy scavenging for energy efficiency in networks and applications,” *Bell Labs Technical Journal*, vol. 15, pp. 7–29, Sept 2010.

[105] B. S. Lee, J. J. He, W. J. Wu, and W. P. Shih, “Mems power receiver using piezoelectric cantilever beams and interdigitated electrodes,” in *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3227–3232, Oct 2006.

[106] J.-M. Dilhac, R. Monthéard, M. Bafleur, V. Boitier, P. Durand-Estèbe, and P. Tounsi, “Implementation of thermoelectric generators in airlinersfor powering battery-free wireless sensor networks,” *Journal of Electronic Materials*, vol. 43, no. 6, pp. 2444–2451, 2014.

[107] J. Vican, B. Gajdeczko, F. Dryer, D. Milius, I. Aksay, and R. Yetter, “Development of a microreactor as a thermal source for microelectromechanical systems power generation,” *Proceedings of the Combustion Institute*, vol. 29, no. 1, pp. 909–916, 2002.

[108] S. Reis, M. Silva, N. Castro, V. Correia, J. Rocha, P. Martins, A. Lasheras, J. Gutierrez, and S. Lanceros-Mendez, “Electronic optimization for an energy harvesting system based on magnetoelectric metglas/poly (vinylidene fluoride)/metglas composites,” *Smart Materials and Structures*, vol. 25, no. 8, p. 085028, 2016.

[109] F.-G. Yuan, *Structural Health Monitoring (SHM) in Aerospace Structures*. Woodhead Publishing, 2016.

[110] S. Zhang and A. Seyedi, “Harvesting resource allocation in energy harvesting wireless sensor networks,” *arXiv preprint arXiv:1306.4997*, 2013.

[111] R. Citroni, A. Leggieri, D. Passi, F. Di Paolo, and A. Di Carlo, “Nano energy harvesting with plasmonic nano-antennas: A review of mid-ir rectenna and application,” *Advanced Electromagnetics*, vol. 6, no. 2, pp. 1–13, 2017.

[112] O. Younis and S. Fahmy, “Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on mobile computing*, no. 4, pp. 366–379, 2004.

[113] J. Bustos-Jiménez, R. Alonso, C. Faúndez, and H. Méric, “Boxing experience: Measuring qos and qoe of multimedia streaming using ns3, lxc and vlc,” in *39th Annual IEEE Conference on Local Computer Networks Workshops*, pp. 658–662, Sept 2014.

[114] B. Bede and L. Stefanini, “Generalized differentiability of fuzzy-valued functions,” *Fuzzy Sets and Systems*, vol. 230, pp. 119–141, 2013.

[115] Y.-J. Liu and S. Tong, “Adaptive fuzzy control for a class of unknown nonlinear dynamical systems,” *Fuzzy Sets and Systems*, vol. 263, pp. 49–70, 2015.

[116] Y. Li, S. Tong, and T. Li, “Adaptive fuzzy output-feedback control for output constrained nonlinear systems in the presence of input saturation,” *Fuzzy Sets and Systems*, vol. 248, pp. 138–155, 2014.

[117] M. Shen and D. Ye, “Improved fuzzy control design for nonlinear markovian-jump systems with incomplete transition descriptions,” *Fuzzy Sets and Systems*, vol. 217, pp. 80–95, 2013.

[118] M. Di and E. M. Joo, “A survey of machine learning in wireless sensor netoworks from networking and application perspectives,” in *2007 6th international conference on information, communications & signal processing*, pp. 1–5, IEEE, 2007.

[119] Y. Wang, M. Martonosi, and L.-S. Peh, “A supervised learning approach for routing optimizations in wireless sensor networks,” in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pp. 79–86, 2006.

[120] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[121] T. Issariyakul and E. Hossain, “Introduction to network simulator 2 (ns2),” in *Introduction to network simulator NS2*, pp. 1–18, Springer, 2009.

[122] G. F. Riley and T. R. Henderson, “The ns-3 network simulator,” *Modeling and tools for network simulation*, pp. 15–34, 2010.

[123] W. Whitt, “A review ofl= λw and extensions,” *Queueing Systems*, vol. 9, no. 3, pp. 235–268, 1991.

[124] L. T. Corporation, “Ltc3588-1 - nanopower energy harvesting power supply,” 2010.

Appendix A: NS-3 c++ code

```
double maxEnergyPiezo = 0.0044;
double minEnergyPiezo = 0.000;

double rEnergy = 10.0;
double maxEnergyLevel = 0.15;
double minEnergyLevel = 0.11;
double iniEnergyLevel = 0.13;
int minQSize = 200;
int maxQSize = 300;

std::ostringstream piezoEnergy;

Ptr<DeviceEnergyModel> basicRadioModelPtr;
Ptr<BasicEnergySource> basicSourcePtr;

//EnergyHarvesterContainer harvesters;

Timer timer (Timer::CANCEL_ON_DESTROY);

NS_LOG_COMPONENT_DEFINE ("EnergyWithHarvestingExample");

std::queue<int> rQueue;

static inline std::string
PrintReceivedPacket (Address& from)
{
  InetSocketAddress iaddr = InetSocketAddress::ConvertFrom (from);

  std::ostringstream oss;
  oss << "--\nReceived one packet! Socket: " << iaddr.GetIpv4 ()
      << " port: " << iaddr.GetPort ()
      << " at time = " << Simulator::Now ().GetSeconds ()
      << "\n--";

  return oss.str ();
}

void
QueuePacket(int packetID)
{
int qSize=maxQSize;
int i = rQueue.size();
if (i <= qSize){
rQueue.push(packetID);
}else{

std::cout << Simulator::Now ().GetSeconds ()
            << " Dropped packet " << packetID  << std::endl;
}
}

int
QueueState( int qSize)
{

int queueState = 0;
```

```
if (qSize <= minQSize )
{
queueState = 0;

}
else if (qSize > minQSize){

queueState = 1;
}



else{

}

return queueState;

}


void
QueueSize()
{
int sz = rQueue.size();


std::cout << Simulator::Now ().GetSeconds ()
            << " Queue Size " << sz  << " " << QueueState(sz)  << std::endl;

}

void
DequeuePacket()
{

if (!rQueue.empty()){

std::cout << Simulator::Now ().GetSeconds ()
            << " Popped packet " << rQueue.front()  << std::endl;
rQueue.pop();
QueueSize();
timer.Schedule ();

}
}

/**
 * \param socket Pointer to socket.
 *
 * Packet receiving sink.
 */
void
ReceivePacket (Ptr<Socket> socket)
{
  Ptr<Packet> packet;
  Address from;
  while ((packet = socket->RecvFrom (from)))
    {
      if (packet->GetSize () > 0)
        {
       QueuePacket(packet->GetUid());
          //NS_LOG_UNCOND (PrintReceivedPacket (from));
       std::cout << Simulator::Now ().GetSeconds ()
                   << " Received one packet " << packet->GetUid()  << std::endl;
        }
    }
}
```

```
/**
 * \param socket Pointer to socket.
 * \param pktSize Packet size.
 * \param n Pointer to node.
 * \param pktCount Number of packets to generate.
 * \param pktInterval Packet sending interval.
 *
 * Traffic generator.
 */
static void
GenerateTraffic (Ptr<Socket> socket, uint32_t pktSize, Ptr<Node> n,
                 uint32_t pktCount, Time pktInterval)
{
  if (pktCount > 0)
    {
      socket->Send (Create<Packet> (pktSize));
      Simulator::Schedule (pktInterval, &GenerateTraffic, socket, pktSize, n,
                           pktCount - 1, pktInterval);
    }
  else
    {
      socket->Close ();
    }
}

void
PutRadiotoIdleandSleep()
{

  std::cout << Simulator::Now ().GetSeconds ()
            << " Radio State " << "0"  << std::endl;

   basicRadioModelPtr->SetAttribute("RxCurrentA", DoubleValue (0.0));
   basicRadioModelPtr->SetAttribute("TxCurrentA", DoubleValue (0.0));


}

void
PutRadiotoActive()
{

  std::cout << Simulator::Now ().GetSeconds ()
            << " Radio State " << "1"  << std::endl;


    basicRadioModelPtr->SetAttribute("RxCurrentA", DoubleValue (0.0174));
    basicRadioModelPtr->SetAttribute("TxCurrentA", DoubleValue (0.0197));


}

double
radioTransProb(int x, int y){

double transProb[3][3] =
            {{0.1, 0.1, 0.1},
{0.1,0.1,0.1},
{0.1,0.1,0.2}};
return transProb[x][y];
}

double
CurrEnergyTransProb(int x, int y){

double transProb[3][3] =
            {{0.1, 0.1, 0.1},
{0.1,0.1,0.1},
{0.1,0.1,0.2}};
```

```
return transProb[x][y];
}


double
HarvestedTransProb(int x, int y){

double transProb[3][3] =
            {{0.1, 0.1, 0.1},
{0.1,0.1,0.1},
{0.1,0.1,0.2}};
return transProb[x][y];
}

int
HarvestedState( double harvestedPower)
{

int harvestState = 0;

if (harvestedPower >= minEnergyPiezo && harvestedPower <= 0.0033)
{
harvestState = 0;
}
else if (harvestedPower > 0.0033 && harvestedPower <= maxEnergyPiezo){

harvestState = 1;
}
else {

}

return harvestState;

}

int
CurrentEnergyState( double currentEnergy)
{

int energyState = 0;

if (currentEnergy <= minEnergyLevel )
{
energyState = 0;
PutRadiotoIdleandSleep();
//piezoEnergy << "ns3::UniformRandomVariable[Min="<<0.003<<"|Max="<<0.008<<"]";
//std::string pe;
//pe = piezoEnergy.str();
//harvesters.SetAttribute("HarvestablePower", StringValue (pe));
}
else if (currentEnergy > minEnergyLevel){

energyState = 1;
PutRadiotoActive();

}

else{

}

return energyState;

}

void
HandleEnergyRecharged(){
```

```
}

int
RadioState()
{
int radioState = 0;

return radioState;
}

void
HandleEnergyDepleted(){

}

void
setMaxEnergyLevel()
{
}

/// Trace function for remaining energy at node.
void
RemainingEnergy (double oldValue, double remainingEnergy)
{
  std::cout << Simulator::Now ().GetSeconds ()
            << "ms Current remaining energy = " << remainingEnergy << "J" <<  " " << CurrentEnergyState(remainingEn

}

/// Trace function for total energy consumption at node.
void
TotalEnergy (double oldValue, double totalEnergy)
{
  std::cout << Simulator::Now ().GetSeconds ()
            << "ms Totalradio energy consumed by radio = " << totalEnergy << "J" << std::endl;
  std::cout << Simulator::Now ().GetSeconds ()
              << "ms Currentradio energy consumed by radio = " << totalEnergy-oldValue << "J" << std::endl;
}

/// Trace function for the power harvested by the energy harvester.
void
HarvestedPower (double oldValue, double harvestedPower)
{
  std::cout << Simulator::Now ().GetSeconds ()
            << "ms Current harvested power = " << harvestedPower << "W" << " " << HarvestedState(harvestedPower) <<


}

/// Trace function for the total energy harvested by the node.
void
TotalEnergyHarvested (double oldValue, double TotalEnergyHarvested)
{
  std::cout << Simulator::Now ().GetSeconds ()
            << "ms Total energy harvested by harvester = "
            << TotalEnergyHarvested << " J" << std::endl;


}







int
main (int argc, char *argv[])
{
```

```
  //LogComponentEnable ("EnergySource", LOG_LEVEL_DEBUG);
 // LogComponentEnable ("BasicEnergySource", LOG_LEVEL_DEBUG);
 // LogComponentEnable ("DeviceEnergyModel", LOG_LEVEL_DEBUG);
 // LogComponentEnable ("RadioEnergyModel", LOG_LEVEL_DEBUG);
  //LogComponentEnable ("EnergyHarvester", LOG_LEVEL_DEBUG);
  //LogComponentEnable ("BasicEnergyHarvester", LOG_LEVEL_DEBUG);



timer.SetFunction(&DequeuePacket);
timer.SetDelay(MilliSeconds(60));
timer.Schedule ();

  std::string phyMode ("DsssRate1Mbps");
  double Prss = -80;            // dBm
  uint32_t PpacketSize = 200;   // bytes
  bool verbose = false;

  // simulation parameters
  uint32_t numPackets = 1000000;  // number of packets to send
  double interval = 0.02;          // seconds
  double startTime = 0.0;       // seconds
  double distanceToRx = 100.0;  // meters
  /*
   * This is a magic number used to set the transmit power, based on other
   * configuration.
   */
  double offset = 81;

  // Energy Harvester variables
  double harvestingUpdateInterval = 1.0;  // seconds

  CommandLine cmd;
  cmd.AddValue ("phyMode", "Phy mode", phyMode);
  cmd.AddValue ("Prss", "Intended primary RSS (dBm)", Prss);
  cmd.AddValue ("PpacketSize", "size of application packet sent", PpacketSize);
  cmd.AddValue ("numPackets", "Total number of packets to send", numPackets);
  cmd.AddValue ("startTime", "Simulation start time", startTime);
  cmd.AddValue ("distanceToRx", "X-Axis distance between nodes", distanceToRx);
  cmd.AddValue ("verbose", "Turn on all device log components", verbose);
  cmd.Parse (argc, argv);

  // Convert to time object
  Time interPacketInterval = Seconds (interval);

  // disable fragmentation for frames below 2200 bytes
  Config::SetDefault ("ns3::RemoteStationManager::FragmentationThreshold",
                      StringValue ("2200"));
  // turn off RTS/CTS for frames below 2200 bytes
  Config::SetDefault ("ns3::RemoteStationManager::RtsCtsThreshold",
                      StringValue ("2200"));
  // Fix non-unicast data rate to be the same as that of unicast
  Config::SetDefault ("ns3::RemoteStationManager::NonUnicastMode",
                      StringValue (phyMode));

  NodeContainer c;
  c.Create (10);     // create 2 nodes
  NodeContainer networkNodes;
  networkNodes.Add (c.Get (0));
  networkNodes.Add (c.Get (1));

  // The below set of helpers will help us to put together the  NICs we want
  Helper ;
  if (verbose)
    {
      .EnableLogComponents ();
    }
  .SetStandard (_PHY_STANDARD_80211b);
```

```
/**  PHY **/
/*****************************************************************************/
YansPhyHelper Phy = YansPhyHelper::Default ();
Phy.Set ("RxGain", DoubleValue (-10));
Phy.Set ("TxGain", DoubleValue (offset + Prss));
Phy.Set ("CcaMode1Threshold", DoubleValue (0.0));
/*****************************************************************************/

/**  channel **/
YansChannelHelper Channel;
Channel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
Channel.AddPropagationLoss ("ns3::FriisPropagationLossModel");
// create  channel
Ptr<YansChannel> ChannelPtr = Channel.Create ();
Phy.SetChannel (ChannelPtr);

/** MAC layer **/
// Add a non-QoS upper MAC, and disable rate control
NqosMacHelper Mac = NqosMacHelper::Default ();
.SetRemoteStationManager ("ns3::ConstantRateManager", "DataMode",
                              StringValue (phyMode), "ControlMode",
                              StringValue (phyMode));
// Set it to ad-hoc mode
Mac.SetType ("ns3::AdhocMac");

/** install PHY + MAC **/
NetDeviceContainer devices = .Install (Phy, Mac, networkNodes);

/** mobility **/
MobilityHelper mobility;
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector (0.0, 0.0, 0.0));
positionAlloc->Add (Vector (2 * distanceToRx, 0.0, 0.0));
mobility.SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (c);

/** Energy Model **/
/*****************************************************************************/
/* energy source */
BasicEnergySourceHelper basicSourceHelper;
// configure energy source
//basicSourceHelper.Set ("BasicEnergySourceInitialEnergyJ", DoubleValue (1.0));
basicSourceHelper.Set ("BasicEnergySourceInitialEnergyJ", DoubleValue (iniEnergyLevel));
basicSourceHelper.Set ("BasicEnergyLowBatteryThreshold", DoubleValue (minEnergyLevel));
basicSourceHelper.Set ("BasicEnergyHighBatteryThreshold", DoubleValue (maxEnergyLevel));
// install source
EnergySourceContainer sources = basicSourceHelper.Install (c);
/* device energy model */
RadioEnergyModelHelper radioEnergyHelper;

// configure radio energy model

  radioEnergyHelper.Set ("TxCurrentA", DoubleValue (0.0174));
  radioEnergyHelper.Set ("RxCurrentA", DoubleValue (0.0197));

  radioEnergyHelper.Set ("IdleCurrentA", DoubleValue (426e-6));
  radioEnergyHelper.Set ("SleepCurrentA", DoubleValue (320e-6));



     piezoEnergy << "ns3::UniformRandomVariable[Min="<<minEnergyPiezo<<"|Max="<<maxEnergyPiezo<<"]";

     std::string pe = piezoEnergy.str();

// install device model
DeviceEnergyModelContainer deviceModels = radioEnergyHelper.Install (devices, sources);

/* energy harvester */
BasicEnergyHarvesterHelper basicHarvesterHelper;
```

```
  // configure energy harvester
  basicHarvesterHelper.Set ("PeriodicHarvestedPowerUpdateInterval", TimeValue (Seconds (harvestingUpdateInterval)))
  //basicHarvesterHelper.Set ("HarvestablePower", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=0.1]"));
  //basicHarvesterHelper.Set ("HarvestablePower", StringValue ("ns3::UniformRandomVariable[Min=0.0022|Max=0.0044]"))
  basicHarvesterHelper.Set ("HarvestablePower", StringValue (pe));
  // install harvester on all energy sources
  EnergyHarvesterContainer harvesters = basicHarvesterHelper.Install (sources);

  /********************************************************************/

  /** Internet stack **/
  InternetStackHelper internet;
  internet.Install (networkNodes);

  Ipv4AddressHelper ipv4;
  NS_LOG_INFO ("Assign IP Addresses.");
  ipv4.SetBase ("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainer i = ipv4.Assign (devices);

  TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory");
  Ptr<Socket> recvSink = Socket::CreateSocket (networkNodes.Get (1), tid);  // node 1, Destination
  InetSocketAddress local = InetSocketAddress (Ipv4Address::GetAny (), 80);
  recvSink->Bind (local);
  recvSink->SetRecvCallback (MakeCallback (&ReceivePacket));recvSink->GetRxAvailable();

  Ptr<Socket> source = Socket::CreateSocket (networkNodes.Get (0), tid);    // node 0, Source
  InetSocketAddress remote = InetSocketAddress (Ipv4Address::GetBroadcast (), 80);
  source->SetAllowBroadcast (true);
  source->Connect (remote);

  /** connect trace sources **/
  /********************************************************************/
  // all traces are connected to node 1 (Destination)
  // energy source
  //Ptr<BasicEnergySource> basicSourcePtr = DynamicCast<BasicEnergySource> (sources.Get (1));
  basicSourcePtr = DynamicCast<BasicEnergySource> (sources.Get (1));
  basicSourcePtr->TraceConnectWithoutContext ("RemainingEnergy", MakeCallback (&RemainingEnergy));
  // device energy model
  //Ptr<DeviceEnergyModel> basicRadioModelPtr =
  basicRadioModelPtr =
    basicSourcePtr->FindDeviceEnergyModels ("ns3::RadioEnergyModel").Get (0);
  NS_ASSERT (basicRadioModelPtr != 0);
  basicRadioModelPtr->TraceConnectWithoutContext ("TotalEnergyConsumption", MakeCallback (&TotalEnergy));

  // energy harvester
  Ptr<BasicEnergyHarvester> basicHarvesterPtr = DynamicCast<BasicEnergyHarvester> (harvesters.Get (1));
  basicHarvesterPtr->TraceConnectWithoutContext ("HarvestedPower", MakeCallback (&HarvestedPower));
  basicHarvesterPtr->TraceConnectWithoutContext ("TotalEnergyHarvested", MakeCallback (&TotalEnergyHarvested));

  /********************************************************************/

  /** simulation setup **/
  // start traffic
  Simulator::Schedule (Seconds (startTime), &GenerateTraffic, source, PpacketSize,
                       networkNodes.Get (0), numPackets, interPacketInterval);

  Simulator::Stop (Seconds (600.0));
  Simulator::Run ();
  Simulator::Destroy ();

  return 0;
}
```

Appendix B: Piezo LTSpice Processing Script

```
PIEZODIR=$HOME/piezoltspice
test -e $PIEZODIR || mkdir $PIEZODIR
./waf --run piezo-ltspice > $PIEZODIR/piezo.txt
cd ~/piezoltspice

grep "Current remaining" piezo.txt | sed 's/J//g' | sed 's/ms//g' > currenergy.data
gnuplot gnu-currenergy.gnu
ps2pdf -dEPSCrop currenergy.eps

gnuplot gnu-currenergy_state.gnu
ps2pdf -dEPSCrop currenergystate.eps

grep "Current harvested" piezo.txt | sed 's/W//g' | sed 's/ms//g' > currharvest.data
gnuplot gnu-currharvest.gnu
ps2pdf -dEPSCrop currharvest.eps

gnuplot gnu-currharvest_state.gnu
ps2pdf -dEPSCrop currharveststate.eps

grep "Currentradio" piezo.txt | sed 's/J//g' | sed 's/ms//g' > currradio.data
gnuplot gnu-currradio.gnu
ps2pdf -dEPSCrop currradio.eps

grep "Totalradio" piezo.txt | sed 's/J//g' | sed 's/ms//g' > totalradio.data
gnuplot gnu-totalradio.gnu
ps2pdf -dEPSCrop totalradio.eps

grep "Radio State" piezo.txt  > radiostate.data
gnuplot gnu-radiostate.gnu
ps2pdf -dEPSCrop radiostate.eps

grep "Queue Size" piezo.txt  > queuesize.data
gnuplot gnu-queuesize.gnu
ps2pdf -dEPSCrop queuesize.eps

gnuplot gnu-queuestate.gnu
ps2pdf -dEPSCrop queuestate.eps

grep "Popped" piezo.txt  > serviced.data
gnuplot gnu-serviced.gnu
ps2pdf -dEPSCrop serviced.eps

grep "Drop" piezo.txt  > dropped.data
gnuplot gnu-dropped.gnu
ps2pdf -dEPSCrop dropped.eps

grep "Received" piezo.txt  > arrivals.data
gnuplot gnu-arrivals.gnu
ps2pdf -dEPSCrop arrivals.eps


awk '{
 rstate[i++] = $4;
 }
 END{
 sum=0
```

```
 s00 = 0
 s01 = 0
 s10 = 0
 s11 = 0
 for (j=0;j<i-1;j++){
  if ((rstate[j] == 0) && (rstate[j+1] == 0)){
   s00 = s00 +1;
   }
  if ((rstate[j] == 0) && (rstate[j+1] == 1)){
   s01 = s01 +1;
   }
  if ((rstate[j] == 1) && (rstate[j+1] == 0)){
   s10 = s10 +1;
   }
  if ((rstate[j] == 1) && (rstate[j+1] == 1)){
   s11 = s11 +1;
   }
 }
 sum = s00+s01+s10+s11
 print "Radio State  "
 print "               " s00/sum "   " s01/sum
 print "               " s10/sum "   " s11/sum
 }
 ' radiostate.data

echo $'\n'

awk '{
 estate[i++] = $7;
 }
 END{
 sum=0
 s00 = 0
 s01 = 0
 s10 = 0
 s11 = 0
 for (j=0;j<i-1;j++){
  if ((estate[j] == 0) && (estate[j+1] == 0)){
   s00 = s00 +1;
   }
  if ((estate[j] == 0) && (estate[j+1] == 1)){
   s01 = s01 +1;
   }
  if ((estate[j] == 1) && (estate[j+1] == 0)){
   s10 = s10 +1;
   }
  if ((estate[j] == 1) && (estate[j+1] == 1)){
   s11 = s11 +1;
   }
 }
 sum = s00+s01+s10+s11
 print "Current Energy State  "
 print "               " s00/sum "   " s01/sum
 print "               " s10/sum "   " s11/sum
 }
 ' currenergy.data

echo $'\n'

awk '{
 hstate[i++] = $7;
 }
 END{
 sum=0
 s00 = 0
 s01 = 0
 s10 = 0
 s11 = 0
 for (j=0;j<i-1;j++){
  if ((hstate[j] == 0) && (hstate[j+1] == 0)){
```

```
   s00 = s00 +1;
   }
  if ((hstate[j] == 0) && (hstate[j+1] == 1)){
   s01 = s01 +1;
   }
  if ((hstate[j] == 1) && (hstate[j+1] == 0)){
   s10 = s10 +1;
   }
  if ((hstate[j] == 1) && (hstate[j+1] == 1)){
   s11 = s11 +1;
   }
 }
 sum = s00+s01+s10+s11
 print "Current Harvest State  "
 print "              " s00/sum "   " s01/sum
 print "              " s10/sum "   " s11/sum
 }
 ' currharvest.data

echo $'\n'

awk '{
 qstate[i++] = $5;
 }
 END{
 sum=0
 s00 = 0
 s01 = 0
 s10 = 0
 s11 = 0
 for (j=0;j<i-1;j++){
  if ((qstate[j] == 0) && (qstate[j+1] == 0)){
   s00 = s00 +1;
   }
  if ((qstate[j] == 0) && (qstate[j+1] == 1)){
   s01 = s01 +1;
   }
  if ((qstate[j] == 1) && (qstate[j+1] == 0)){
   s10 = s10 +1;
   }
  if ((qstate[j] == 1) && (qstate[j+1] == 1)){
   s11 = s11 +1;
   }
 }
 sum = s00+s01+s10+s11
 print "Queue State  "
 print "              " s00/sum "   " s01/sum
 print "              " s10/sum "   " s11/sum
 }
 ' queuesize.data

echo $'\n'


awk '{
 serv[i++] = $1;
 }
 END{
 sum=0
 for (j=0;j<i-1;j++){
 sum=sum+serv[j+1]-serv[j]
 }
 print "Service rate: " i/sum
 }
 ' serviced.data

echo $'\n'

awk '{
 arrv[i++] = $1;
```

```
 }
 END{
 sum=0
 for (j=0;j<i-1;j++){
 sum=sum+arrv[j+1]-arrv[j]
 }
 print "Arrival rate: " i/sum
 }
 ' arrivals.data

echo $'\n'

awk '{
 qs[i++] = $4;
 }
 END{
 sum=0
 for (j=0;j<i;j++){
 sum=sum+qs[j]
 }
 print "Avrg Queue size: " sum/i
 }
 ' queuesize.data

echo $'\n'


awk '(NR == FNR){qtime1[i++] = $1; packetid1[x++] = $5;next}
{packetid2[y++] = $4; qtime2[j++] = $1}
 END{
 sum=0
counter=0
 for (n=0;n<x;n++){
 for (m=0;m<y;m++){
 if (packetid2[m] == packetid1[n])
{

 sum=sum+qtime2[m]-qtime1[n]
}
 }
 }
 print "Total queue time : " sum
 }
 ' arrivals.data serviced.data

echo $'\n'
```