UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

PROJECT TITLE

AGENTS BASED CONTROL AND COORDINATION MODEL FOR A DISTRIBUTED WIRELESS SENSOR NETWORK TO MAXIMIZE FIELD LIFE.

BY

KIRUI, RONALD KIPYEGON

SUPERVISOR: ERIC .M. AYIENGA

Submitted in partial fulfillment of the requirement for the award of Masters of Science Degree in Computer Science.

DECLARATION

This project, as presented in this report, is my original work and has not been presented for any other award in any other University.

NAME: KIRUI RONALD KIPYEGON

REG. NO: P58/70504/2008

Signature: ____________________

Date: 19/11/2012

This project has been submitted as partial fulfillment of the requirements for the degree of Master of Science in Computer Science of the University of Nairobi with my approval as the University supervisor.

Name: **Mr. ERIC AYIENGA**

Signature: ____________________

Date: 19/11/2012

TABLE OF CONTENTS

ABSTRACT

In recent years, wireless sensor network deployments for real life applications have rapidly increased. However, energy control remains one of the major barriers hindering complete exploitation of this technology because sensor nodes are usually powered by batteries with a limited lifetime. Therefore efficient energy management is key requirement for the success of a wireless sensor network. In this project, we proposed a distributed agent based model to solve the issue of energy management in WSNs. In our approach, sensor nodes are modeled as agents. One of the agents is a controller agent called the expert sensor agent (ESA). The ESA at the start of sensing sets a threshold energy level. During sensing, the ESA continuously monitors each of the other sensor agents' energy levels and if a sensor node's power level falls below the threshold, the ESA deactivates the node and at the same time activates one of the nodes currently in sleep mode. At the of the day, the sensing work load is equitably distributed across the nodes. The model was implemented using simulation experiments in OPNET. We evaluated our agent based model against none agent based technique in terms of power consumed and end to end delay (ETE). Our simulation results indicate that our agent based model achieves over 75% power savings compared to the non-agent techniques while at the same time ensuring real time delivery of data.

LIST OF ABBREVIATIONS

WSN - Wireless Sensor network <put here some brief definition>

DSR - Dynamic source routing , <put here some brief definition>

MAC - medium Access Control., <put here some brief definition>

SA - Sensor Agent, <put here some brief definition>

EA - Expert Agent, <put here some brief definition>

CSIP - Collaborative Signal and Information processing, <put here some brief definition>

DCHS - Distributed constraint heuristic search , <put here some brief definition>

MAS - Multiagent system, <put here some brief definition>

DWSN - Distributed wireless sensor network, <put here some brief definition>

ETE Delay – End to End delay, <put here some brief definition>

ACKNOWLEDGEMENT

Thanks to all people who provided guidance to me in producing this project. In particular, I convey thanks to my supervisor Mr. Eric Ayienga from The University of Nairobi, School of computing and Informatics for giving me insight on the workings of this Master's project, and for the many discussions on the topic, especially for providing guidance and his flexibility to modify my project in the directions as required and more so his availability whenever requested. I also give my special thanks to my family members i.e. my spouse and my children for giving me moral support and humble time towards the success of my project.

I must also pass my appreciation to my panel team members Mr. Tonny Onwansa and Dr. Elisha Opiyo who guided and gave me direction in full collaboration with my supervisor to ensure a success in my project.

Finally, I convey thanks to my class colleagues and all those who encouraged me along the way.

CHAPTER ONE: INTRODUCTION

1.1 Background

Wireless Sensor Networks (WSNs) is a recent and significant improvement over traditional sensor networks arising from advances in wireless communications, microelectronics and miniaturized sensors. In fact, the low power, multi-functional sensor nodes are tiny in size, have embedded processing as well as communication capabilities.

Recent advances in microprocessor fabrication have led to a dramatic reduction in the size and power consumed by embedded microprocessor-controlled devices. Battery and sensing technology together with radio hardware have also followed a similar miniaturization trend. The aggregation of these advances has led to the development of networked, millimeter-scale, sensing devices capable of complex processing tasks. Collectively these form a Wireless Sensor Network (WSN), which is the technology required to enable a new era for ubiquitous sensing technology.

Large scale deployments of these networks have been used in many diverse fields such as wildlife habitat monitoring, traffic monitoring, lighting control, Wild-fire monitoring, smart farming/harvesting, habitat monitoring, structural health monitoring, surveillance and emergency response systems and Military battle fields.

A distributed sensor network therefore is an autonomous system of thousands of mobile devices connected by wireless links. The devices are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably.

A significant amount of work has already been done in different aspects of wireless sensor network. Most researchers have surveyed a number of such research efforts in wireless sensor network.

The futuristic application scenarios bring out two key requirements of sensor networks i.e. support for very large number of unattended autonomous nodes and adaptability to environment and task dynamics. As more success stories of sensor networks in different application domains are being reported, the number of nodes in a wireless sensor networks is also growing. Similarly, sensor networks are now subjected to perform in extreme environments like forests and vineyards where they come across variations in operating conditions and node failures. Scalability and adaptability are existing challenges in wireless sensor networks without which their application will be severely limited. Sensor nodes in a wireless sensor network almost always operate on battery occasionally backed by solar or wind energy sources. Sensor nodes therefore have to make optimal use of the available energy resources.

Power management is therefore a critical aspect of a successful WSN. This may take many forms. Typically the sensors spend most of their time in a low power sleep mode. Even when active, energy-intensive activities such as using actuators and transmitting/receiving radio messages are kept to a minimum. Hence there is a need to come up with more power managing technique designs that employ agents to make cooperative decisions to reduce the energy consumption of different aspects of the functioning of a typical WSN.

1.2 Problem Statement

Most problems that arise in distributed wireless sensor networks revolve around its requirement to have a computation of a global conclusion that is consistent to local information to each sensor node. Wireless sensor networks have received increasing attention and research in recent years due to their flexibility, cost effectiveness, ease of deployment, scalability and dynamic coverage. Wireless Sensor network consist of many inexpensive wireless sensor nodes each capable of collecting, processing, storing and communicating information (sending data).

Wireless sensor networks however are faced with several challenges such as Ad-hoc deployment, unattended operation, **Power constraints** and Dynamic changes in the environment.

It is important to acknowledge that the success of any wireless sensor network fully relies upon the ability of the sensor node power to sustain the intended application domain. This therefore implies that the sensor battery power level for any wireless sensor network is a critical factor. Hence it is necessary to come up with better ways and strategies to contain the power constraints.

The complexity of overall sensor network operation is increased with the growing sophistication of data fusion techniques, tracking algorithms, sensor device capabilities, network device characteristics and amount of relevant detail in the phenomena being tracked.

Most of the research work on sensor networks focuses on techniques to relay sensed information in an energy-efficient manner to a central base station. In addition, methods for collaborative signal and information processing (CSIP) which attempt to perform processing in a distributed and collaborative manner among several sensor nodes have also been proposed.

As more success stories of sensor networks in different application domains are being reported, the number of nodes in a wireless sensor networks is also growing. Similarly, sensor networks are now subjected to perform in extreme environments like forests and vineyards where they come across variations in operating conditions and node failures. Scalability and adaptability are also existing challenges in wireless sensor networks without which their application will be severely limited. Sensor nodes therefore have to make optimal use of the available energy resources.

The major portion of the energy budget in a sensor node is spent on transmission and reception of the sensor data. It is therefore possible to minimize communication related energy usage in a sensor node by using a suitable communication protocol and several such algorithms have already been proposed. There has been a detailed research and surveys of such protocols specifically designed for sensor networks where energy awareness is an essential consideration.

Most known power aware communication protocols follow a cluster based approach in which a group of nodes in a region select a cluster head (CH) that gather the information from nodes in the cluster and forward it to the sink(base station). The most interesting research issue regarding such protocols is how to form the clusters so that the energy consumption and contemporary communication metrics such as latency are optimized.

But due to the limited power supply of these devices, it is imperative that the operation of each node factor need to be taken into account when making any decision to perform an action. For example, we can consider the case where a nodes power has depleted to a point where it is capable of only a handful of transmissions. In such a situation the network should be able to adapt to this fact and the node should only transmit if it deems it critical to do so. This decision must be taken on a per node basis as off node deliberation would necessitate further transmissions. Intelligent and proactive behavior such as this is a major characteristic of an intelligent software agent.

1.3 *Objectives of this project*

1.3.1 Research Objectives

To justify how agent technology can be used to minimize energy constraints on sensor nodes hence maximizing field life of a wireless sensor network deployment based on demonstrations results achieved via simulations.

1.3.2 System Development Objectives

- To develop an agent based model to demonstrate how energy savings can be achieved in a distributed wireless sensor network with the ultimate goal of maximizing the field life.

1.4 Research Questions and Expected Outcome

- Does the agent based model improve the overall wireless sensor network deployed field life and guarantee tangible results as per its anticipated design?

1.5 Research Context and Setup

This research is aimed at developing an agent based model that will be used to demonstrate how agents can aid in minimizing power loss in a wireless sensor node by using energy management based algorithms.

The overall idea behind this project is to try and identify the mechanism over which Distributed constrained heuristic search (DCHS) algorithm is used in a wireless sensor network to identify their potential in the presence of agents.
The other main issue is to ascertain the ability of sensors node to sustain their energy levels upon the presence of agents on the distributed wireless network.

The key concept behind this project is to try and identify with the use of a distributed heuristics constraint search algorithm which employs dynamic source routing with the ability of achieving substantial power savings by controlling and coordinating the usual sensor node energy level with agents support and hence guaranteeing the actual field life deployment duration.

1.6 Audience of this Research

The target group for this research are financial institutions such as banks, Telecommunication firms, Wildlife management service, security based agencies (Police, NSIS, or security firms) where security violation is a major drawbacks in relation to their set out regulations.

These groups could gain by adopting this model to ascertain the most suitable wireless sensor network deployment mechanism or systems based on their area of application based on the evaluation outcomes.

CHAPTER TWO : LITRATURE REVIEW

There is currently growing research on the field of agents based approach in trying to provide solutions to various problems encountered in wireless sensor networks.

The coordination of sensors by running sensors as agents helps to realize mission-based tasks. Allowing sensors to be shared and reassigned between different tasks which also increase power savings.

To verify this reality various researchers have been trying to come up with various strategies with the aim of solving the problem of sensor power constraints in regards to wireless sensor network domain.

[Akkaya, 2005] studied the approaches on data routing in sensor networks and classified the approaches into three main categories, namely data-centric, hierarchical and location based. In their classification of routing protocols in wireless sensor networks they indicated the routing protocols which utilizes data aggregation will achieve energy saving and traffic optimization. They neither established nor provided which of the algorithms is more energy efficient and/or will prolong the lifetime of a wireless

[Qi, 2003] proposed the mobile agent based distributed sensor network (MADSN) for scalable and energy efficient data aggregation by transmitting the software code, called a mobile agent (MA), to sensor nodes, a large amount of sensory data can be reduced or transformed into a small amount of data by eliminating the redundancy. However, the operation of an MADSN is based on the following assumptions: (1) the sensor network architecture is clustering based; (2) each source node is within one hop from a cluster head; (3) much redundancy exists among the sensory data which can be fused into a single data packet with a fixed size.

These assumptions pose much limitation on the range of applications that can be supported by an MADSN.

[J Kho, 2010] developed two different decentralized algorithms to solve the problem of communication in wireless visual sensor networks where each node acts as an agent. In the first algorithm (Fixed Routing), route from nodes to the base station is not changed. In the second algorithm, route between nodes and base station is not fixed. To sense, route and forward messages, three types of messages are used. First type of message contains node sampling data. Other messages called coordination messages are meta-data messages and control messages. Meta-data messages include content of visual data with the number of samples during production of the data. Control messages allocate resources to supply efficient energy consumption. Sending control messages before actual data messages increases necessary data rate for the base station. In fixed-routing, there is only one predefined unique path between a node and the base station. In flexible-routing, an arbitrary path has been selected for routing the data by the nodes. There are levels for each node to indicate the distance between node and the base station. Nodes belong to the third level have 3-hop distance to the base station. Each node sends its data to a node that is nearer to the base 26 station. So a tree-structured routing mechanism has been constructed in the network. Transmission, routing costs, and process time during sampling have been compared between these approaches. Flexible routing transmits data twice according to fixed routing. However, communication and computation during sampling costs are 100 times more for flexible routing. The more nodes the network has, the more efficiency obtained by the fixed routing.

[Chansu Yu, 2003] Surveyed and classified the energy aware routing protocols proposed for mobile ad-hoc networks. In their study they demonstrated using energy aware routing protocols minimize either the active communication energy required to transmit or receive packets using transmission power control approach and load distribution approach. They showed that in many cases, it is difficult to compare these routing protocols directly since each method has a different goal with different assumptions and employs different means to achieve the goal. They concluded that more research is needed to combine and integrate some of the protocols presented to keep wireless sensor networks functioning for a longer duration.

[Vasu, 2006], did an excellent work in defining data compression techniques e.g. coding by ordering, pipelined in-network compression and data reduction techniques e.g. information dissemination via label forwarding, differential coding, which can be used to reduce power consumption during routing, but in the end they did not provide which routing algorithm will result into more network lifetime using those t to the developers to apply any algorithm. This means if an algorithm is not suitable for optimal network lifetime then the gains from those data compression and reduction techniques may be eroded.

[G Anastasi, 2008] surveyed strategies for reducing the power consumption acting at the radio level. They presented a systematic and comprehensive taxonomy of the energy conservation schemes. However they sited on their observations about the different approaches to energy management that as far as "traditional" techniques to energy saving and an important aspect which they claim has to be investigated more deeply is the integration of the different approaches into a single off-the-shelf workable solution. This they stressed the need of involving characterizing of the interactions between different protocols and exploiting cross-layer interactions.

[Shah, 2003] proposed to use a set of sub-optimal paths occasionally to increase the lifetime of the network. These paths are chosen by means of a probability function, which depends on the energy consumption of each path. Network survivability is the main metric that the approach is concerned with. The approach argues that using the minimum energy path all the time will deplete the energy of nodes on that path. Instead, one of the multiple paths is used with a certain probability so that the whole network lifetime increases. The protocol assumes that each node is addressable through a class-based addressing which includes the location and types of the nodes.

In [Woo and Zhao, 2004] the volatility in link quality in wireless sensor networks is studied. Zhao shows the existence of "gray areas" where some nodes exceed 90% successful reception while neighboring nodes receive less than 50% of the packets. He shows that the gray area is rather large–one-third of the total communication range. Woo independently verified Zhao's gray area findings.

In designing a reliable multi-hop routing protocol, Woo shows that effectively estimating link qualities is essential. Snooping on traffic over the broadcast medium is crucial for extracting information about the surrounding topology. By snooping, network protocols can prevent cycles, notify neighboring nodes of unreachable routes, improve collision avoidance, and provide link quality information. Since data must ultimately be reported out of the network, the media access protocol must be flexible to meet changing network protocol demands.

CHAPTER THREE : ANALYSIS AND DESIGN

3.1 Research Methodology

Multi-agent systems are a new paradigm for understanding and building distributed systems, where it is assumed that the computational components are autonomous, that is ability to control their own behavior in the furtherance of their own goals.

Agent-based systems technology has been hailed as a new paradigm for conceptualizing, designing, and implementing software systems which are distributed in nature. Agents are sophisticated computer programs that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems. Increasingly, however, applications require multiple agents that can work together. In particular, multi-agents have shown their potential to meet critical needs in high-speed, mission-critical, content-rich, distributed information systems where mutual interdependencies, dynamic environments, uncertainty, and sophisticated control play a notable role.

There exist several agent methodologies that can be used in designing and developing a multiagent based system. Some of the common agent methodologies include:

- **Australian AI Institute(AAII) agent development methodology**

  This is where internal and external models are used. External model identifies the agents and their interactions. Internal models represent the implementations.

- **The TROPOS agent methodology**

  This methodology has four phases, which are the early requirements phase, late requirement phase, architectural design phase and detailed design phase.

- **MAS-CommonKADS methodology**

This is an agent-oriented software engineering methodology that guides the process of analyzing and designing multi-agent systems.

- **Process for Agent Societies Specification and Implementation(PASSI) methodology**

  This is a step-by-step requirement to-code methodology for: designing and developing multi-agent societies integrating design models and concepts from both object-oriented (OO) software engineering and MAS, using the Unified Modeling Language (UML) notation.

- **The Gaia Methodology**

  This methodology uses the analogy of human-based organizations.

- **The Prometheus methodology**

This is an iterative methodology covering the complete software engineering process i.e. Analysis, Design, Detailed design, & Implementation

Other emerging uses of multiagent systems are in layered systems architectures in which agents at different layers need to coordinate their decisions to achieve appropriate configurations of resources and computational processing and in the design of survivable systems in which agents dynamically reorganize to respond to changes in resource availability, this is a typical direction for designing a agents based system model for a distributed wireless sensor networks with a view of achieving intended goal (which is maximizing the field life of sensor nodes due to prolonged power savings).

3.2 *Prometheus Methodology*

For this project Ichoose to use Prometheus which is an iterative methodology covering the complete software engineering process i.e. Analysis, Design, Detailed design, & Implementation. It has a complete-lifecycle methodology for analyzing, designing and developing heterogeneous MAS. It uses a goal-driven development and is independent of a particular multi-agent system architecture, programming language, or message passing system; hence it's suitability in designing agent based wireless sensor network deployments experiments.

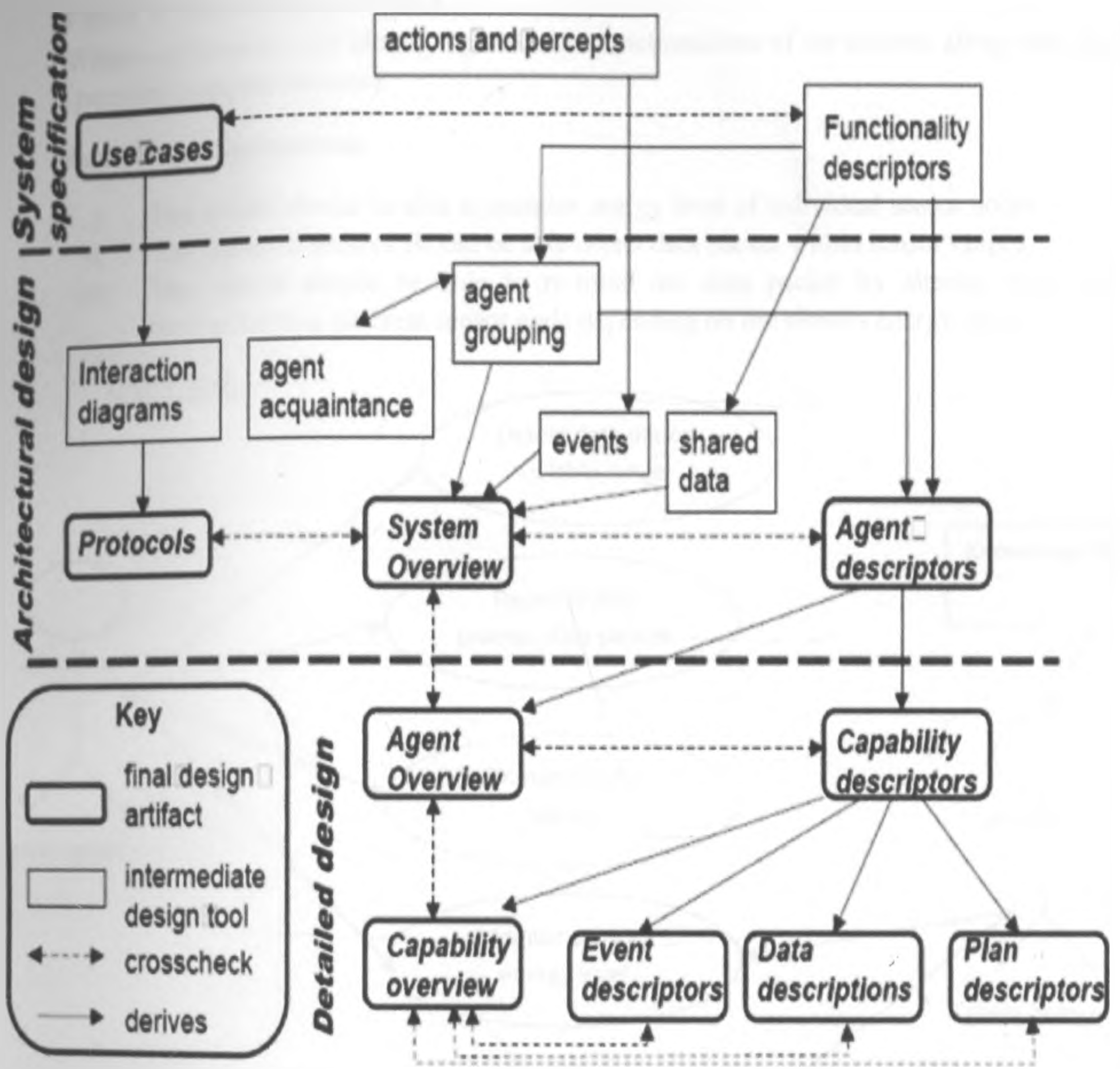
The following activities will be carried out:

- The system specification phase focused on identifying the basic functionalities of the system, along with inputs (percepts), outputs (actions) and any important shared data sources.
- The architectural design phase used the outputs from the previous phase to determine which agents the system will contain and how they will interact.
- The detailed design phase we will look at the internals of each agent and how it will accomplish its tasks within the overall system.

METHODOLOGY DESIGN

The project system design will involve using Prometheus methodology because it is detailed and complete in the sense of covering all activities required in developing intelligent agent systems. Prometheus methodology has complete software engineering process i.e Analysis, Design, Detailed design, & Implementation. It is a complete-lifecycle methodology for analyzing, designing and developing multi-agent applications. It uses a goal-driven development and is independent of a particular multi-agent system architecture, programming language, or message passing system.

3.3 Prometheus Design Phases



The following activities are carried out:

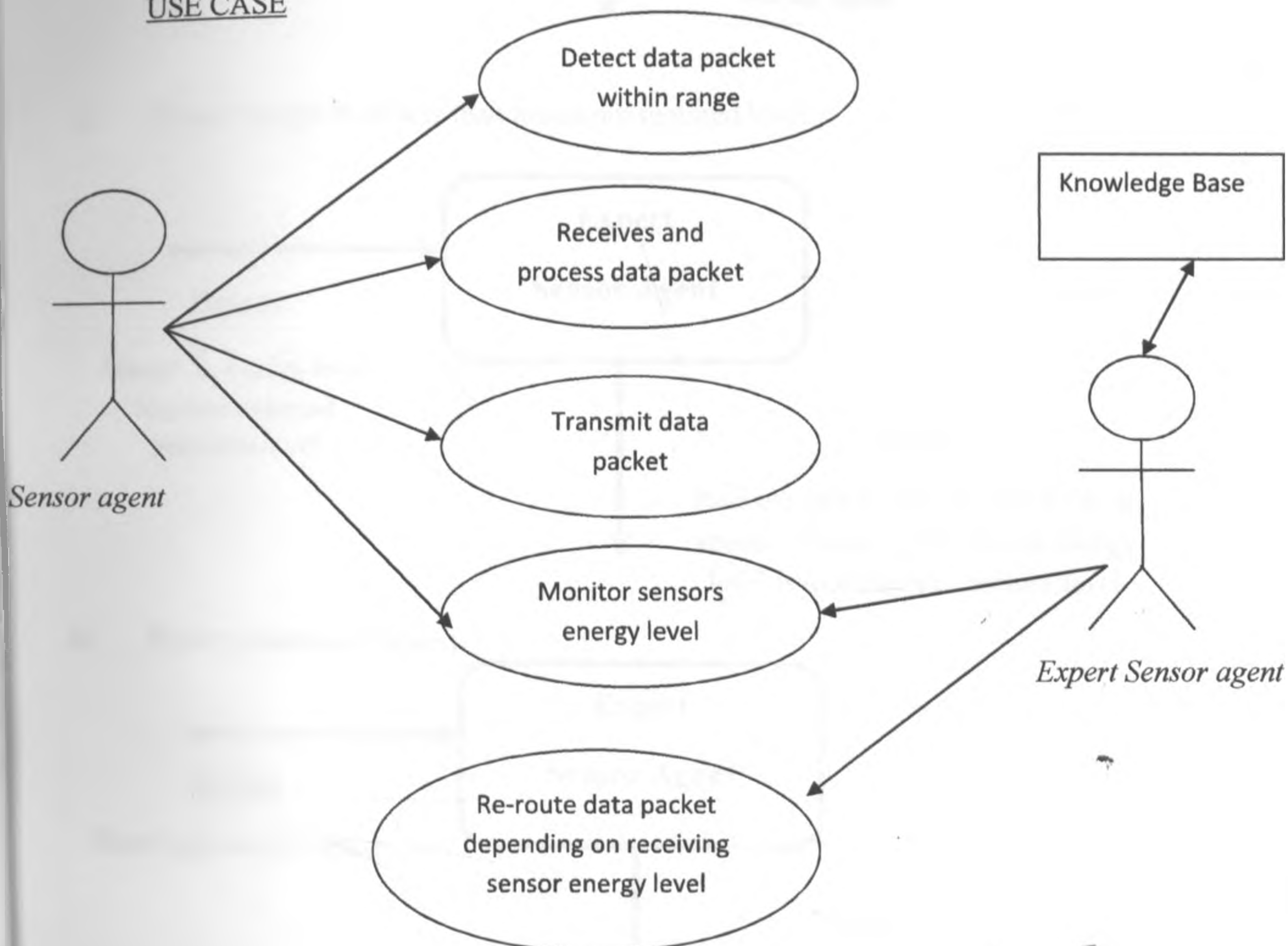
Phase I: System specification

Where the focuses is on identifying the basic functionalities of the system, along with inputs (percepts), outputs (actions).

Functional requirements

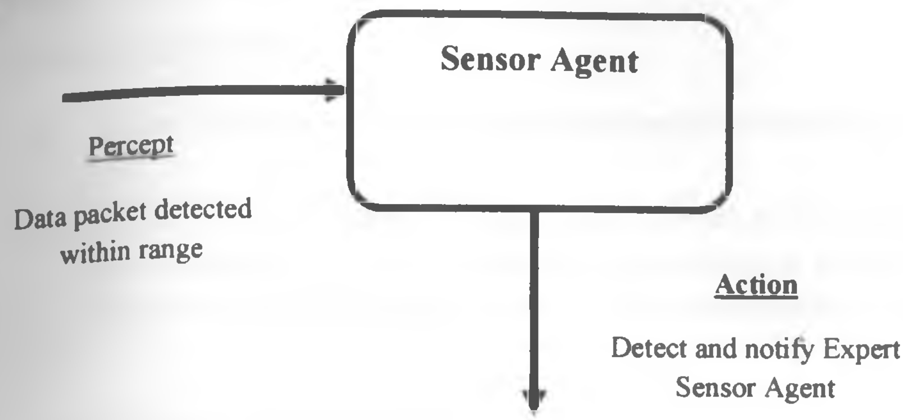
i. The system should be able to monitor energy level of individual sensor nodes
ii. The deployed sensors should be able detect data packet within sensor ranges
iii. The system should be able to re-route the data packet by altering data packet destination to a different sensor node depending on the sensors energy level

USE CASE

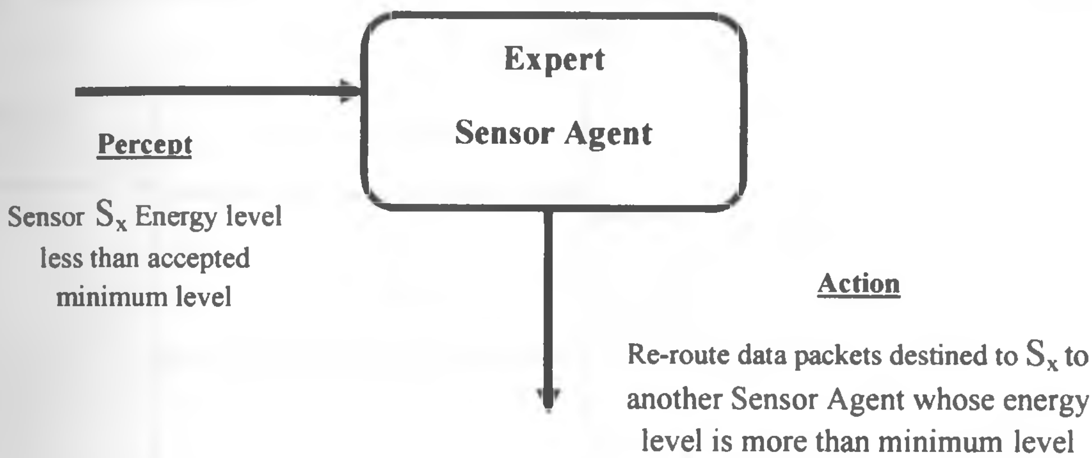


PERCEPTS AND ACTION

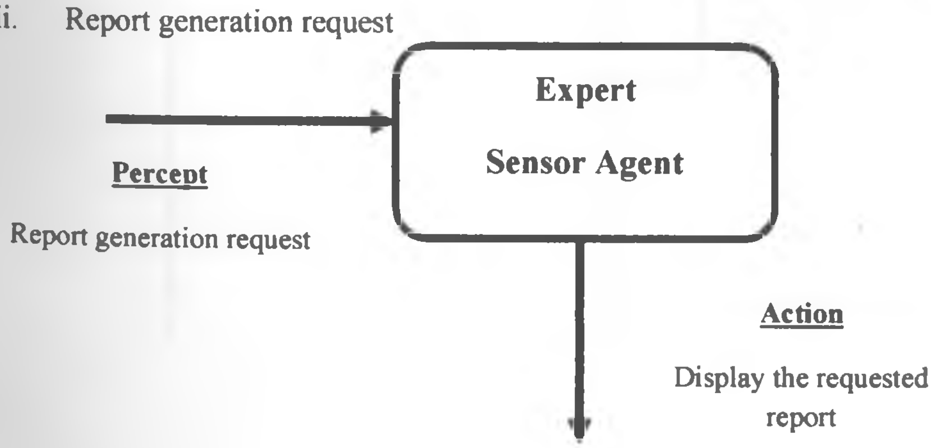
i. Data packet within sensor range



ii. Sensor Energy level less than minimum required level



iii. Report generation request



PHASE II: Architectural design

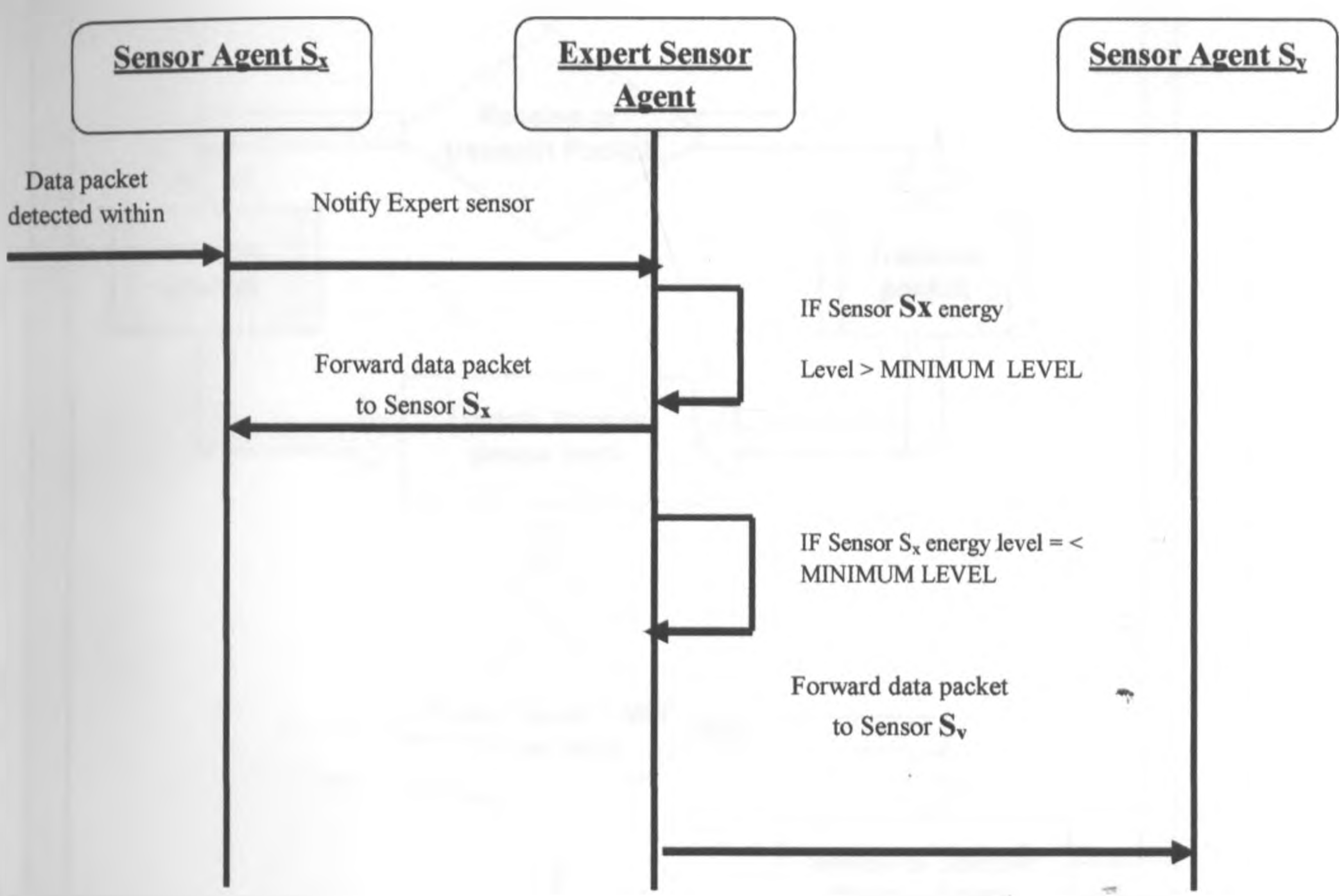
This is a design stage whereby the output from the previous phase is used by the next agent level in the system and show how they interact.

AGENT GROUPING

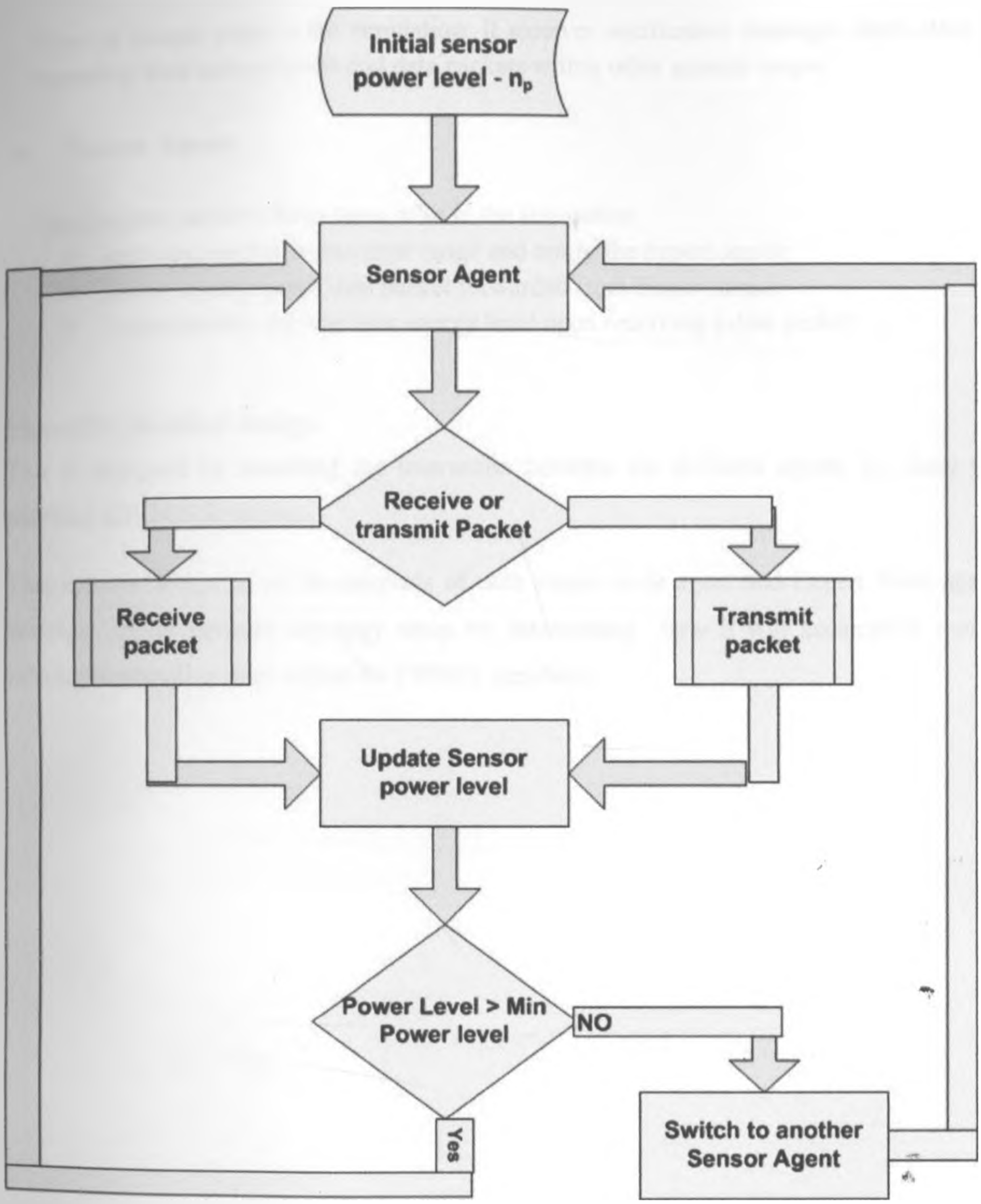
i. **Sensor Agents** – Is a group agents that cannot re-route data packet

ii. **Expert Sensor Agent** – A single agent whose role is to coordinate and monitor other sensors. It has the capability of monitoring other sensors' energy levels and re-routing packets destined to a particular sensor to a different sensor.

INTERRACTIVE DIAGRAM



SYSTEM FLOWCHART



AGENT DESCRIPTION

i. Expert Sensor Agent

Exist as unique node in the simulation. It receives notification messages from other sensors regarding their energy levels and data packets within other sensors ranges.

ii. Sensor Agents

These agents perform three main roles in the simulation:

- etect data packet within their range and notify the expert sensor
- Receive and process data packet forwarded from Expert sensor
- Automatically deplete their energy level upon receiving a data packet

PhaseIII: Detailed design

This is designed by modeling the interaction between the different agents by using **OPNET MODELER 14.5** Simulator.

This involve design of all the internals of each sensor node agent and Expert Node agent for a Wireless sensor network topology setup by determining how it will accomplish their tasks within the overall system within the OPNET simulator.

SENSORS DEPLOYMENT TOPOLOGY

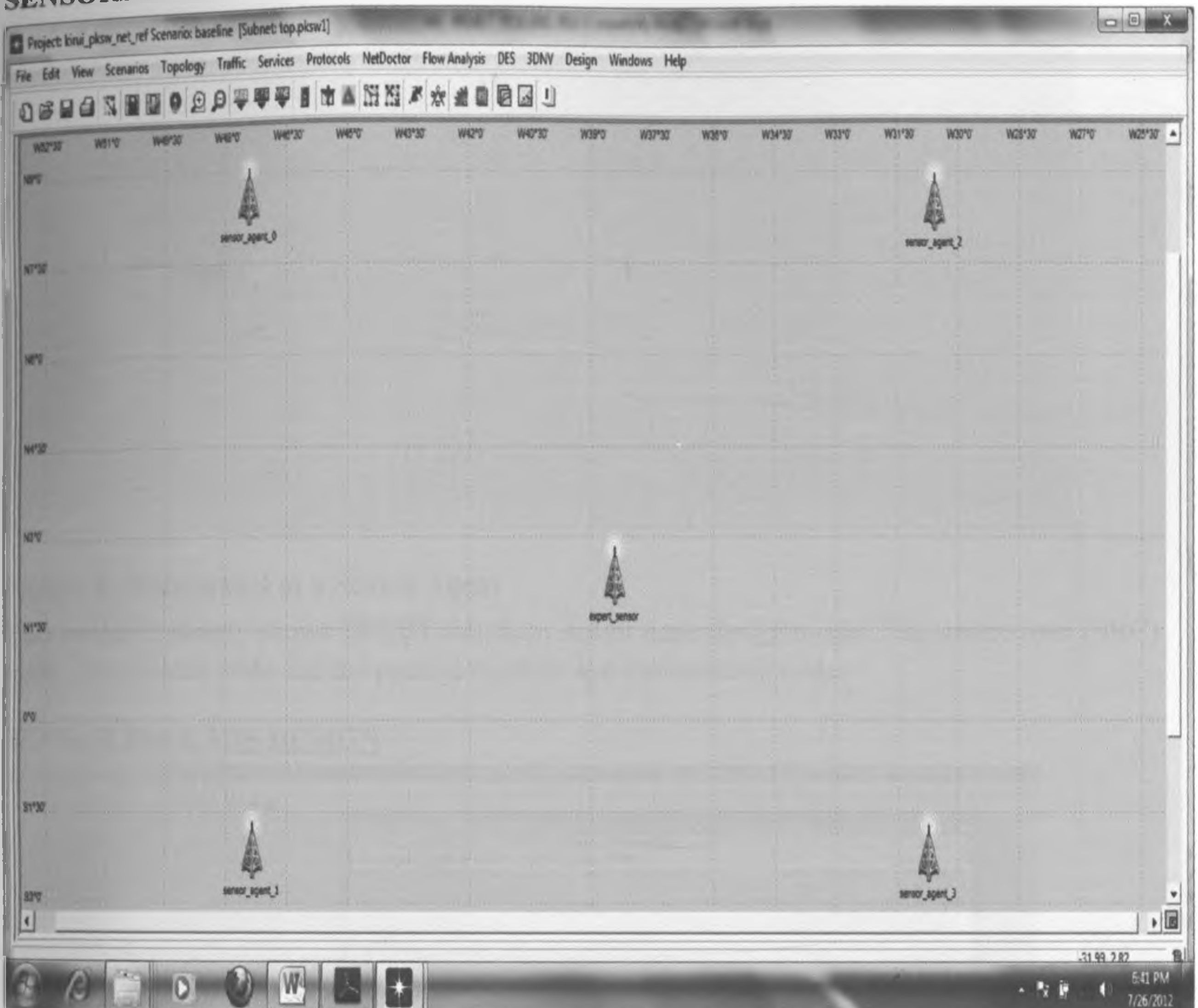


Figure 1

This is OPNET Simulator sensor deployment topology clearly showing how sensor node agents can be distributed within a given environment with an Expert sensor Agent as the controlling and co-rdinating point

SENSOR NODE DESIGN

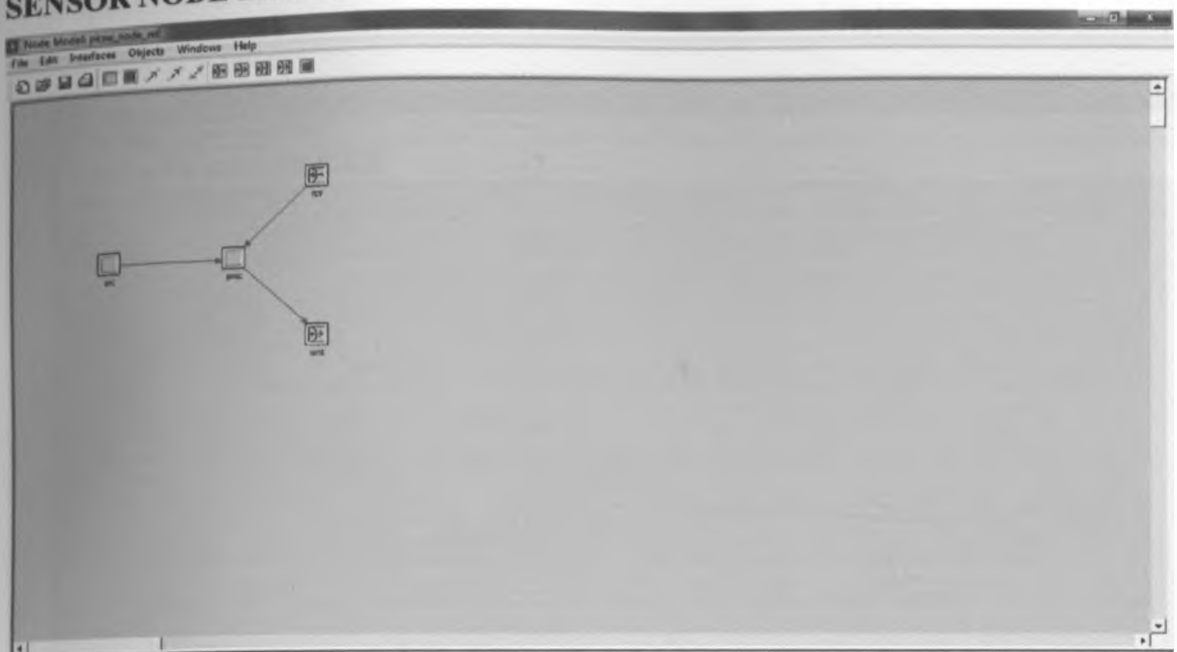


Figure 2: Node model of a Sensor Agent

This snapshot clearly shows OPNET simulator sensor node design model ,The source reset (SRC) node, the process node and the packets receiver and transmission nodes

SENSOR PROCESS DESIGN

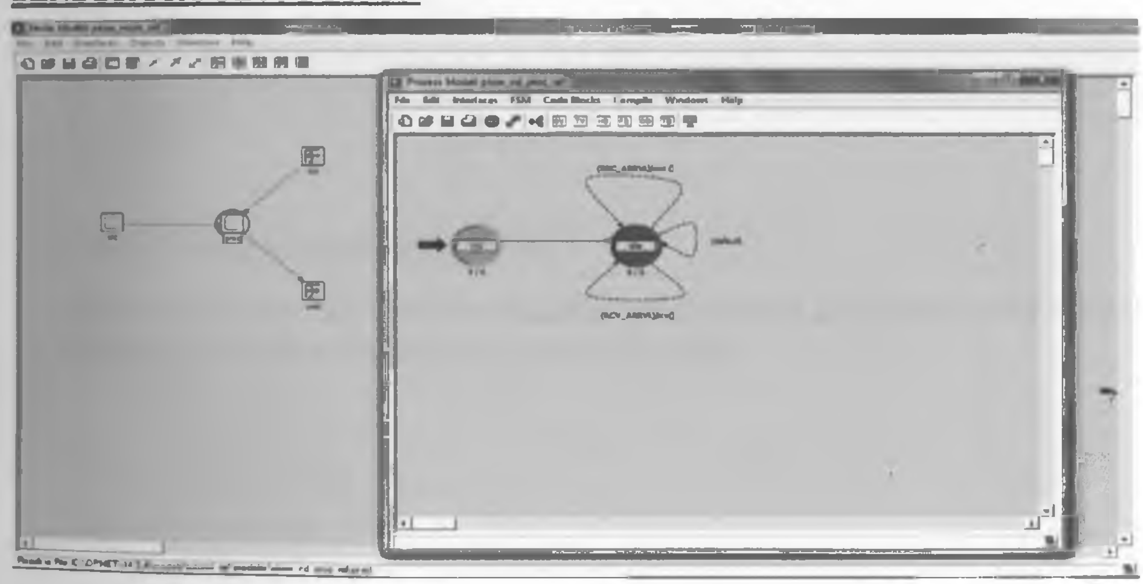


Figure 3. Finite State Machine of a Sensor Agent Node

This snapshot clearly shows a sensor node process model ,The Finite state machine(FSM) shows two sensor node processes which are the initial state and the idle state.

EXPERT SENSOR NODE DESIGN

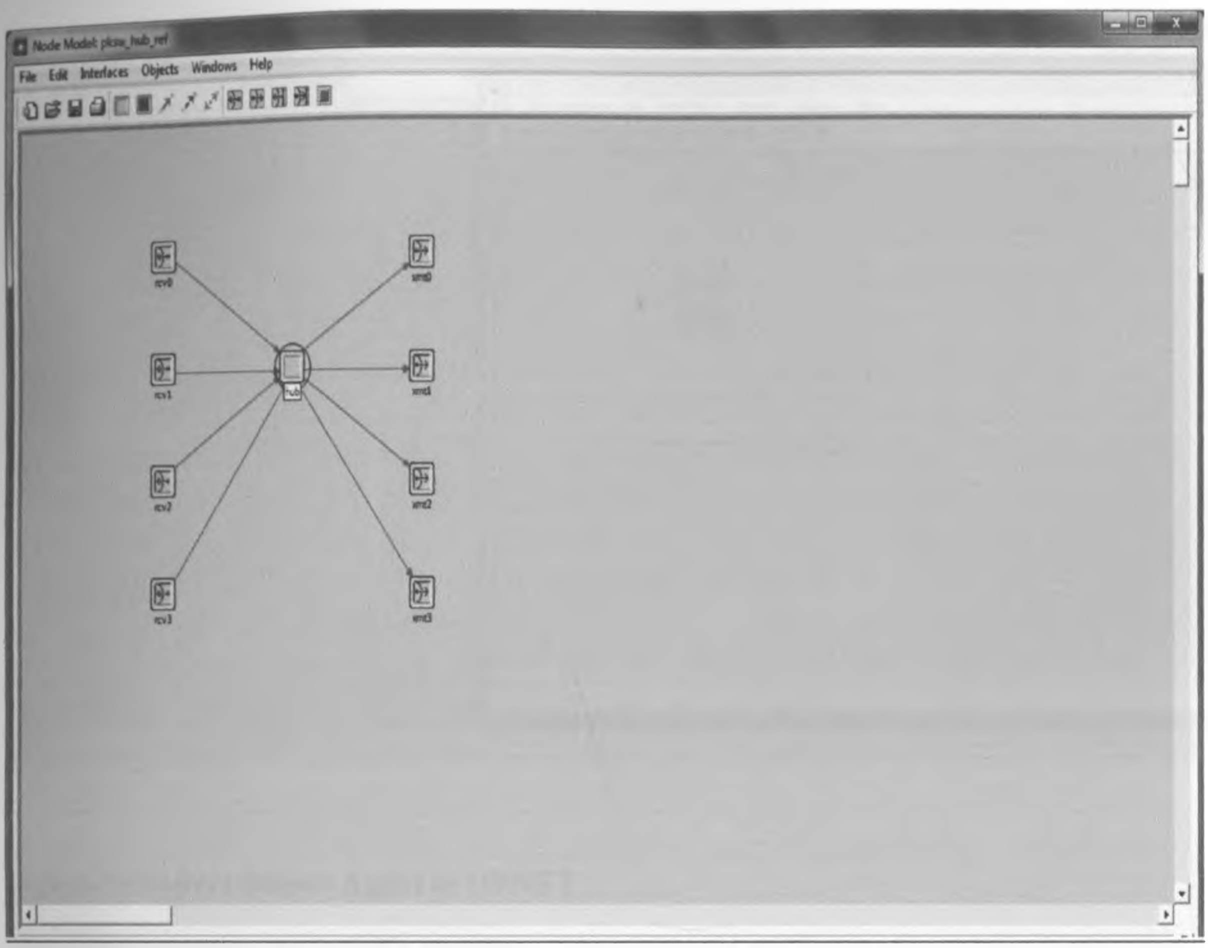


Figure 4: Sensor Agent Node Model

This snapshot from the OPNET simulator clearly shows an Expert sensor node model with packets received and transmitted to the sensor nodes.

EXPERT SENSOR PROCESS DESIGN

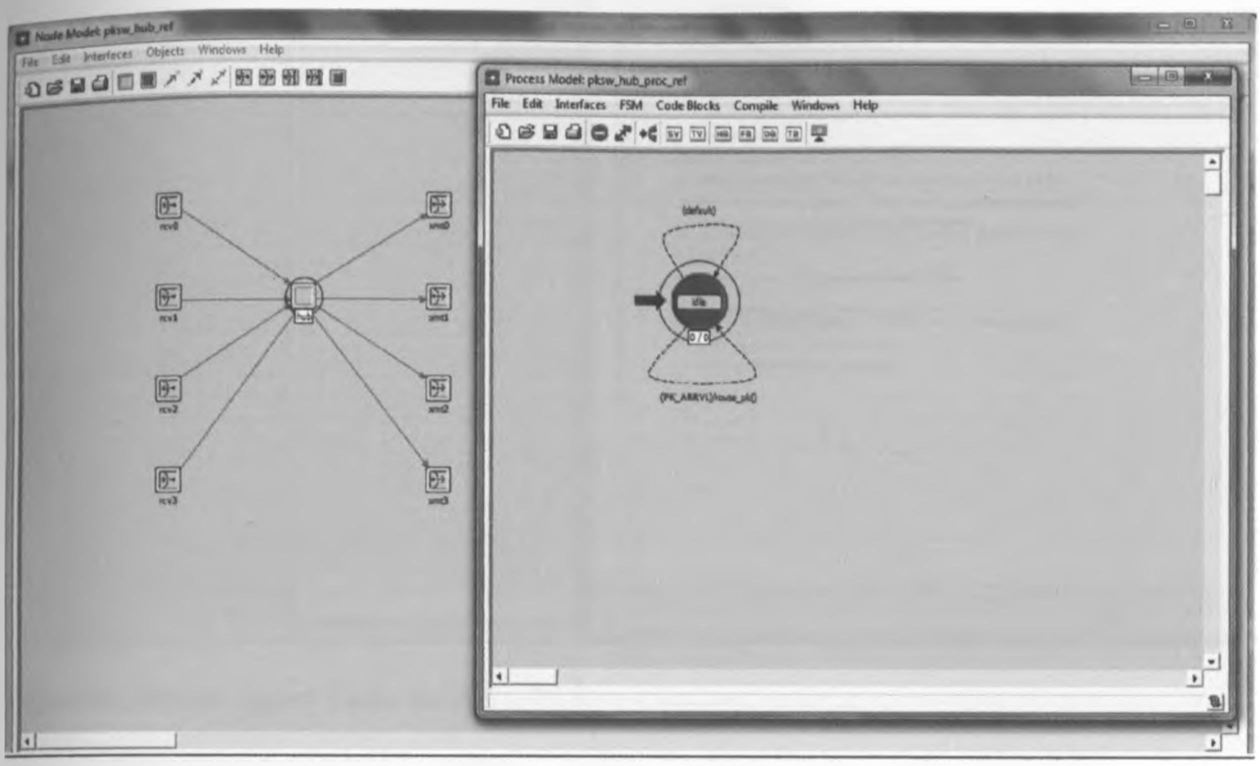


Figure 5: Expert Sensor Agent in OPNET

This snapshot from the OPNET simulator clearly shows an Expert sensor node agent process model .The finite state machine (FSM) shows idle state where all the sensor nodes agents packets default are controlled and coordinated in terms of their power depletion levels.

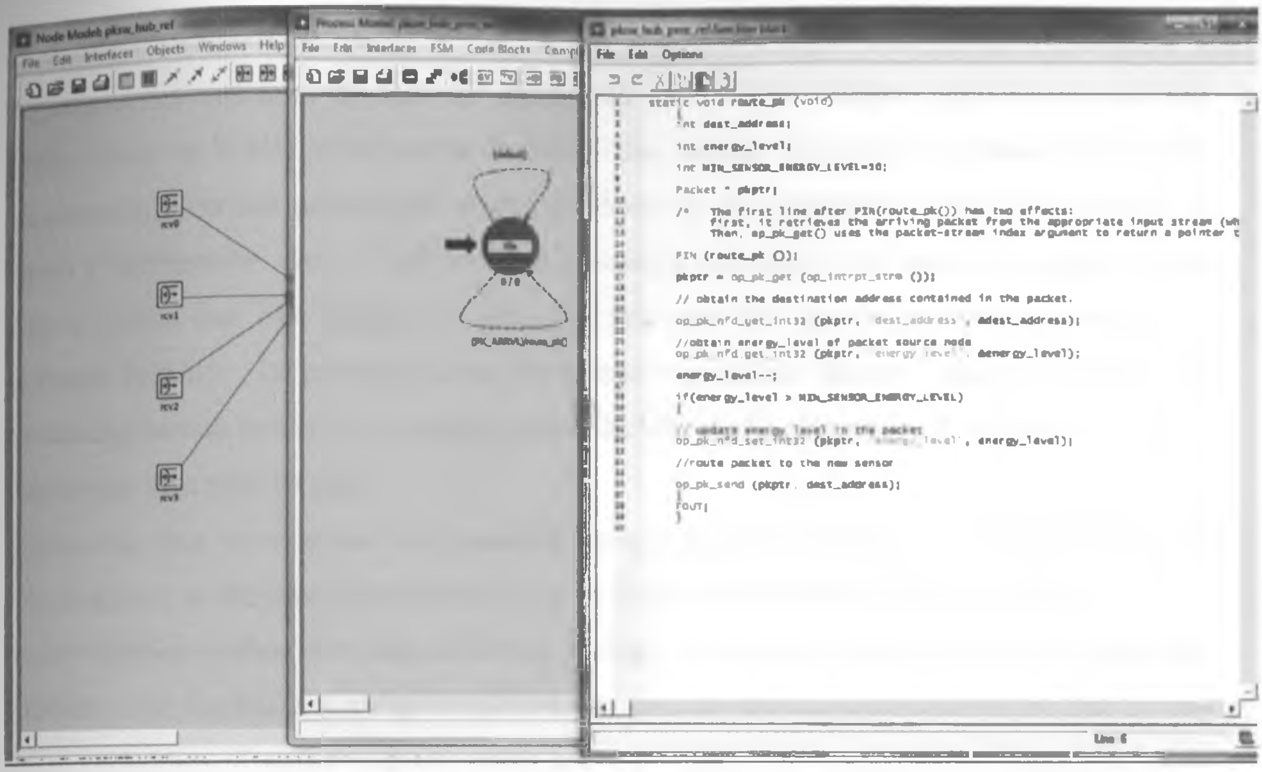


Figure 6: Sensor Agent Code Model

This snapshot from the OPNET simulator clearly shows an Expert sensor node agent process model ,the finite state machine (FSM) state and Function block code.

ASSUMPTIONS

a. Initial sensor nodes energy levels are equal.

b. Sensor Nodes are randomly distributed in a defined region.

c. Energy used in sensing, processing, idle periods is not a bottleneck and thus constant.

d. Sensor Nodes route information towards specified gateway nodes.

e. Mobile targets are randomly generated and move randomly across the plane of the defined region.

3.4 The Proposed Solution

Agents in this case form the basis for distributed, artificially intelligent applications, and their applicability in WSNs is the central theme of this project. Making an informed decision by incorporating various perceptions or beliefs about an environment is at the very heart of an agent's deliberative cycle. Goal oriented reasoning can allow the agent to commit to the course action that best realizes its goals, in this case the agents goal is to maximize the network Feld life . Its perceptions are the sensory modalities that are capable of sensing the remaining battery power. By reasoning about the effect of a transmission, the agent can see if that action best suits its goals.

Following this we examine the potential for using agent technology by applying agents, which adhere to the strong notion of agency, to intelligent decision making in WSNs.

Agent abilities include here data collection, storage, manipulation and autonomous reasoning. Mobility and flexible reasoning in unknown or partially known environment are also among those properties of agent systems which are very useful for usage in the designs of WSN systems.

The solution proposed for this project therefore is to develop an agent based Model by using OPNET MODELER simulator platform to demonstrate how sensor power consumption can be minimized by employing agents for a distributed wireless sensor network. This will entail employing a distributed constraint heuristic search algorithm which in this case uses dynamic source routing (DSR) approach that will perform the control and coordination of sensors with the assistance of agents deployed in a given region set.

The model (Simulator) will be operating at the **Data-link layer** of the OSI model; this will involve the modeling of sensor power depletion against time, control and coordination processes of the agents Simulation of an actual sensor.

The Sensor node (agents) will be running on the MAC layer of the system which will in this case be on the **Network layer** of the OSI Model all based on IEEE 802.15.4 standard.

The network layer in this aspect has two primary functions i.e. routing and addressing of nodes.

Traditional wireless sensor network addressing will assigns fixed addresses to the sensor nodes (agents); the advantage of this scheme is that the addresses will be made unique.

The sensor residual power will be at the physical layer of the OSI reference model.

Communication between sensor nodes agents will require creating a physical link between respective radios. The physical layer will therefore be handling the communication of the sensor node agents across this physical link, which involves modulating and coding the node energy (power) data so that the intended receiver which in this case is the Expert sensor agent can optimally decode it in the presence of channel non-idealities and interference.

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.1 Simulation Results and Analysis

4.1.1 Simulation Results

The simulation results from this project was obtained by using OPNET MODELER 14.5 simulator, In order to test the simulator design, functionality, and usefulness regarding a distributed wireless sensor Network, two main scenarios was used for this experiment.

The first scenario shows a basic simulation design where sensor agent was not active during the simulation runs and the other Simulation run involved a situation where sensor agents was active ,as per Figure1,the Expert sensor Agent was performing the control and coordination of sensor agents for both simulation runs within a DWSN's deployment. As for the simulator itself, the results have shown that agents can substantial solve the power constraints problem encountered on wireless sensor networks deployments.

To analyze my results I used two performance metrics:

- End To End (ETE) Delay
- Power Consumption (in watts)

RESULTS ANALYSIS

As per data and graphs obtained in can be substantially concluded as follows:

ETE Delay Analysis:

In terms **of ETE delay** from the results shown in reference to appendix 1 Figure 7 and Figure 10 graphs

DES simulation Run 1, is a case where sensor agents was not all active,

The ETE delay was at an average of 0.44 units.

While on the other hand DES simulation run 2 was executing at an ETE Delay of an average of 0.33 units.

This shows that the ETE delay for simulation Run 1 was higher with an average of 0.11 units.

It can be clearly summarized that the ETE delay for simulation Run 1 is more by **30 %.**

Power consumption analysis:

This can be clearly deduced from run 1 & run 2 OPNET 14.5 simulator results comparison for (energy consumption vs. Simulation time) line graphs in reference to appendix 1 Figure12,Figure 13,Figure14 and Figure 15 that:

Simulation run 1 power consumption is at an **average of 2.5 Watts**

While Simulation Run 2 consumption is at an **average of 0.5 watts.**

From the above results obtained from the simulation runs, it can be clearly summarized that with the Agents support **over 75%** of sensor node power can be saved on a wireless sensor network.

CHAPTER FIVE: CONCLUSION AND FUTURE WORK

This project main research direction was geared towards extending the lifetime of sensor Node units encompassing using agent technology. The general framework was to perceive sensor nodes as sensor agents whose main role is to regulate sensor node battery energy consumption, where the overall agents control and coordination is being performed by an expert sensor agent, which will replace in this context a role played by a traditional sink (base station)in the current real life scenarios. The ultimate goal of is the minimizing of power consumption on sensor node hence improving their overall field life.

Sensor Node battery longevity has been always sited by several researchers to be critical factor to a success in entire operation of a wireless sensor networks. This research has presented efforts at addressing this very issue through an agent control and coordination oriented approach advocating towards the solution of distributed, contextualized and collaborative power constraint problems in a wireless sensor network set up. Agents have been inherently viewed in the currently existing literature to be suited where the decision making context is highly dynamic, resource bounded and the information is partial and inaccurate.

As with all decisions regarding solving problems encountered on wireless sensor networks, it is important to note that a tradeoff exists between the cost (in terms of node power depletion) that needs not exceed the benefit that can be derived Inevitably on the issue of sensor node power management which is always inextricably associated with other network characteristics and challenges.

The correlation between such attributes as network coverage, latency, accuracy and longevity should be well recognized and be considered in wider perspective while deriving solutions towards the success of any wireless sensor network deployment performance.

Decisions relating to goals of sensor node power longevity therefore must be made mindful of their implications for other network entire characteristics.

As from this perspective, From this project in can be strongly concluded that, Agent technology can be appreciated and be focused in future with a view of being capable of providing solutions to ever dynamic problems encountered on wireless sensor networks such as network coverage, Latency issues ,unattended operations just to mention a few.

REFERENCES

[K. Akkaya and M. Younis, 2005] "A Survey of Routing Protocols in Wireless Sensor Networks," in the Elsevier Ad Hoc Network Journal, Vol. 3/3 pp. 325-349

[A Schumacher,2008]"Lifetime Maximization in Wireless Sensor Networks by Distributed Binary Search," Wireless Sensor Networks, 5th European Conference, EWSN 2008, Bologna, Italy, January 30-February 1, 2008.

[X Wang 2007] INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE Collaborative signal processing for target tracking in distributed wireless sensor networks - Journal of Parallel and Distributed Computing, 2007 – Elsevierhttp://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4215581, last accessed <put date here>

"Center for embedded networked sensing (CENS)," http:// cens.ucla.edu/., last accessed <put date here>

"Picoradio, the berkeley wireless research center picoradio project," http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/.

"Wireless integrated network sensors (WINS)," http://www.janet.ucla.edu/WINS/., last accessed <put date here>

"MIT µAMP (µ-adaptive multi-domain power aware sensors) project," http://www-mtl.mit.edu/research/icsystems/uamps/.

[Yang Ji-yun ,2009] Routing protocol of the wireless sensor networks for bridge health monitoring [J]; Journal of Chongqing University; (College of Computer Science, Chongqing University,Chongqing 400030,P.R. China); 2009-12 http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJC201007008.htm, last accessed <put date here>

[Yuan ,2008] structural health monitoring Wireless Sensor Network(WSN) fault diagnosis fault detection Spatial correlation degree. I Sai1, 2, YUAN Shen-fang2, LI Han-guang1 (1.School of Computer and Software,Nanjing University of Information Science & Technology,Nanjing 10044;2.Aeronautic Science Key Laboratory for Smart Materials & Structures,Nanjing University of Aeronautics and Astronautics,Nanjing 210016) http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJC201007008.htm , last accessed <put date here>

[SUN ,2010] (Institute of Electrical Engineering Yanshan University,Qinhuangdao Hebei 066004,China); An Algorithm of Nodes Scheduling in Coverage Area of Wireless Sensor Networks Clustering Topology [J]; Chinese Journal of Sensors and Actuators; 2010-01. http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJC201007034.htm, last accessed <put date here>

[Akyildiz, 2002] "A survey on sensor networks," IEEE Communications Magazine, 2002.

[F Stajano ,1999] "The resurrecting duckling: Security issues in ad-hoc wireless networks," in Proceedings of the 7th International Workshop on Security Protocols, B. Christianson, B. Crispo, and M. Roe, Eds. 1999, Lecture Notes in Computer Science, Computer Science, and Springer-Verlag.

[Seys and Bart, 2003]"Cost evaluation of efficient digital signature schemes for low Power devices," Internal Report STS–0301, K.U.Leuven, ESAT, SCD/COSIC, 2003.

[Hairong Qi,2003] Journals> Proceedings of the IEEE ...> Volume: 91 Issue: 8
Mobile-agent-based collaborative Signal and information processing in sensor networks; Dept. of Electrical. & Computing Engineering., Univ. of Tennessee, Knoxville, TN, USA (2003).

[Taewook kang,2007], "A Clustering Method for Energy Efficient Routing in Wireless Sensor Networks," Proceedings of the 6th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Corfu Island, Greece, February 16-19, 2007.

[Alessandro Mei,2012] Journal of Parallel and Distributed Computing Volume 72Fine grained load balancing in multi-hop wireless networks Pages 475-488 http://www.sciencedirect.com/science/journal/07437315/72, last accessed <put date here>

[Vasu, 2006] "Energy-Efficient Routing in Wireless Sensor Networks Based on Data Reduction," The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing Las Vegas, Nevada, USA (June 26-29, 2006).

[J Polastre, 2010] "Versatile Low Power Media Access forWireless Sensor Networks" Computer Science Department University of California, Berkeley, CA 94720

[J Kho, 2010] School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. ."An Agent-Based Distributed Coordination Mechanism for Wireless Visual Sensor Nodes Using Dynamic Programming" The Computer Journal, Vol. 53 No. 8

[Chansu Yu, 2003] "Energy efficient routing protocols for mobile Ad-hoc Networks," wireless communication mobile computing; Volume 3: issue 8 Pages 959–973.

[G. Anastasi,2008] "Energy Conservation inWireless Sensor Networks", Ad Hoc Networks, (currently available at http://info.iet.unipi.it/~anastasi/papers/adhoc08.pdf).

[V. Raghunathan, 2006] "Emerging Techniques for Long Lived Wireless SensorNetworks", IEEE Communications Magazine, Vol. 44, N. 4, April 2006.

[Rahul C. Shah and Jan M. Rabaey, 2002]) (Energy Aware Routing for Low Energy Ad-Hoc Sensor Networks) Berkeley Wireless Research Center University of California, Berkeley Volume 1, pp. 17- 21, March 2002.

http://bwrc.eecs.berkeley.edu/publications/2002/presentations/wcnc2002/wcnc.rahul.pdf, last accessed <put a date here>

[Edmund H,1999], Distributed Problem Solving and Planning, 3.4 :Result shairing from the book pp. 133 (Multiagent Systems (A Modern Approach to Distributed Modern Approach to Artificial Intelligence) Edited by Gerhard Weiss The MIT Press Cambridge, Massachusetts London, England)).

CHAPTER SIX:APPENDICES

Appendix :Results

DES RUN 1 SIMULATOR RESULTS DATA

TIME (SEC)	DES-1: ETE DELAY
10.22	0.22
58.33	0.33
98.22	0.22
102.33	0.33
150.33	0.33
154.22	0.22
198.33	0.33
202.22	0.22
206.22	0.22
250.22	0.22
254.22	0.22
302.22	0.22
306.22	0.22
350.22	0.22
354.22	0.22
390.22	0.22
394.33	0.33
450.33	0.33
454.22	0.22
498.22	0.22
502.22	0.22
550.22	0.22
554.22	0.22
598.22	0.22
602.22	0.22
650.22	0.22
654.33	0.33
698.22	0.22
702.22	0.22
750.22	0.22
754.33	0.33
798.22	0.22
802.44	0.44
850.33	0.33
854.22	0.22
898.44	0.44
902.22	0.22
906.33	0.33
950.22	0.22
998.22	0.22

[illegible]	0.44

DES RUN 2 SIMULATOR SIMULATION TIME VS ENERGY CONSUMPTION RESULTS DATA

time (sec)	Kirui_pksw_net_ref-baseline-DES-2: pksw1.expert_sensor <-> sensor agent. Point-to-point. Utilization <--
0	0
10	0.55
20	0.366666667
30	0.275
40	0.22
50	0.366666667
60	0.314285714
70	0.275
80	0.244444444
90	0.33
100	0.3
110	0.275
120	0.253846154
130	0.314285714
140	0.293333333
150	0.275
160	0.258823529
170	0.305555556
180	0.289473684
190	0.275
200	0.261904762
210	0.3
220	0.286956522
230	0.275
240	0.264
250	0.296153846
260	0.285185185
270	0.275
280	0.265517241
290	0.293333333
300	0.283870968
310	0.275

320	0.266666667
330	0.291176471
340	0.282857143
350	0.275
360	0.267567568
370	0.289473684
380	0.282051282
390	0.275
400	0.268292683
410	0.288095238
420	0.281395349
430	0.275
440	0.268888889
450	0.286956522
460	0.280851064
470	0.275
480	0.269387755
490	0.286
500	0.280392157
510	0.275
520	0.269811321
530	0.285185185
540	0.28
550	0.275
560	0.270175439
570	0.284482759
580	0.279661017
590	0.275
600	0.270491803
610	0.283870968
620	0.279365079
630	0.275
640	0.270769231
650	0.283333333
660	0.279104478
670	0.275
680	0.271014493
690	0.282857143
700	0.278873239
710	0.275
720	0.271232877
730	0.282432432

740	0.278666667
750	0.275
760	0.271428571
770	0.282051282
780	0.278481013
790	0.275
800	0.271604938
810	0.281707317
820	0.278313253
830	0.275
840	0.271764706
850	0.281395349
860	0.27816092
870	0.275
880	0.271910112
890	0.281111111
900	0.278021978
910	0.275
920	0.272043011
930	0.280851064
940	0.277894737
950	0.275
960	0.272164948
970	0.280612245
980	0.277777778
990	0.275
1000	#N/A

DES RUN 1 SIMULATOR SIMULATION TIME VS ENERGY CONSUMPTION RESULTS DATA

time (sec)	Kirui_pksw_net_ref-baseline-DES-1: pksw1. Expert sensor <-> sensor agent. Point-to-point. Utilization -->
0	0
10	1.65
20	1.833333333
30	2.2
40	2.2
50	2.383333333
60	2.357142857
70	2.475
80	2.444444444
90	2.53
100	2.5
110	2.566666667
120	2.538461538
130	2.592857143
140	2.566666667
150	2.6125
160	2.588235294
170	2.627777778
180	2.605263158
190	2.64
200	2.619047619
210	2.65
220	2.630434783
230	2.658333333
240	2.64
250	2.665384615
260	2.648148148
270	2.671428571
280	2.655172414
290	2.676666667
300	2.661290323
310	2.68125
320	2.666666667
330	2.685294118
340	2.671428571

350	2.688888889
360	2.675675676
370	2.692105263
380	2.679487179
390	2.695
400	2.682926829
410	2.697619048
420	2.686046512
430	2.7
440	2.688888889
450	2.702173913
460	2.691489362
470	2.704166667
480	2.693877551
490	2.706
500	2.696078431
510	2.707692308
520	2.698113208
530	2.709259259
540	2.7
550	2.710714286
560	2.701754386
570	2.712068966
580	2.703389831
590	2.713333333
600	2.704918033
610	2.714516129
620	2.706349206
630	2.715625
640	2.707692308
650	2.716666667
660	2.708955224
670	2.717647059
680	2.710144928
690	2.718571429
700	2.711267606
710	2.719444444
720	2.712328767
730	2.72027027
740	2.713333333
750	2.721052632
760	2.714285714

770	2.721794872
780	2.715189873
790	2.7225
800	2.716049383
810	2.723170732
820	2.71686747
830	2.723809524
840	2.717647059
850	2.724418605
860	2.718390805
870	2.725
880	2.719101124
890	2.725555556
900	2.71978022
910	2.726086957
920	2.720430108
930	2.726595745
940	2.721052632
950	2.727083333
960	2.721649485
970	2.72755102
980	2.722222222
990	2.728
1000	#N/A

DES RUN 1 SIMULATOR ETE DELAY RESULTS BAR GRAPH

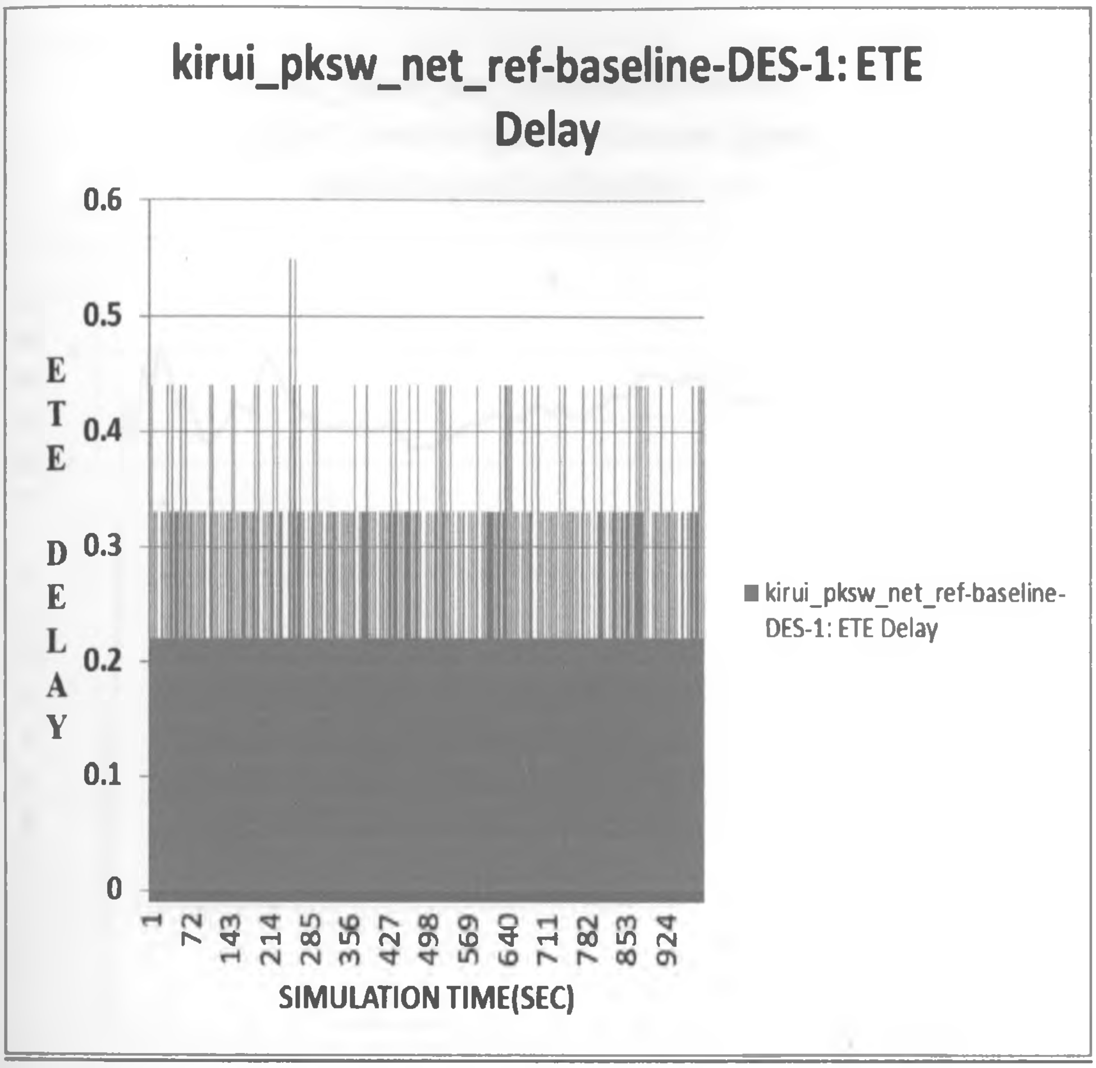


Figure 7

This bar graph clearly shows the discrete event OPNET simulation results for Run 1 in terms of End to end delay performance over 1000 seconds time execution.

RUN 1 SIMULATOR RESULTS(ENERGY CONSUMPTION VS SIM TIME) LINE GRAPH

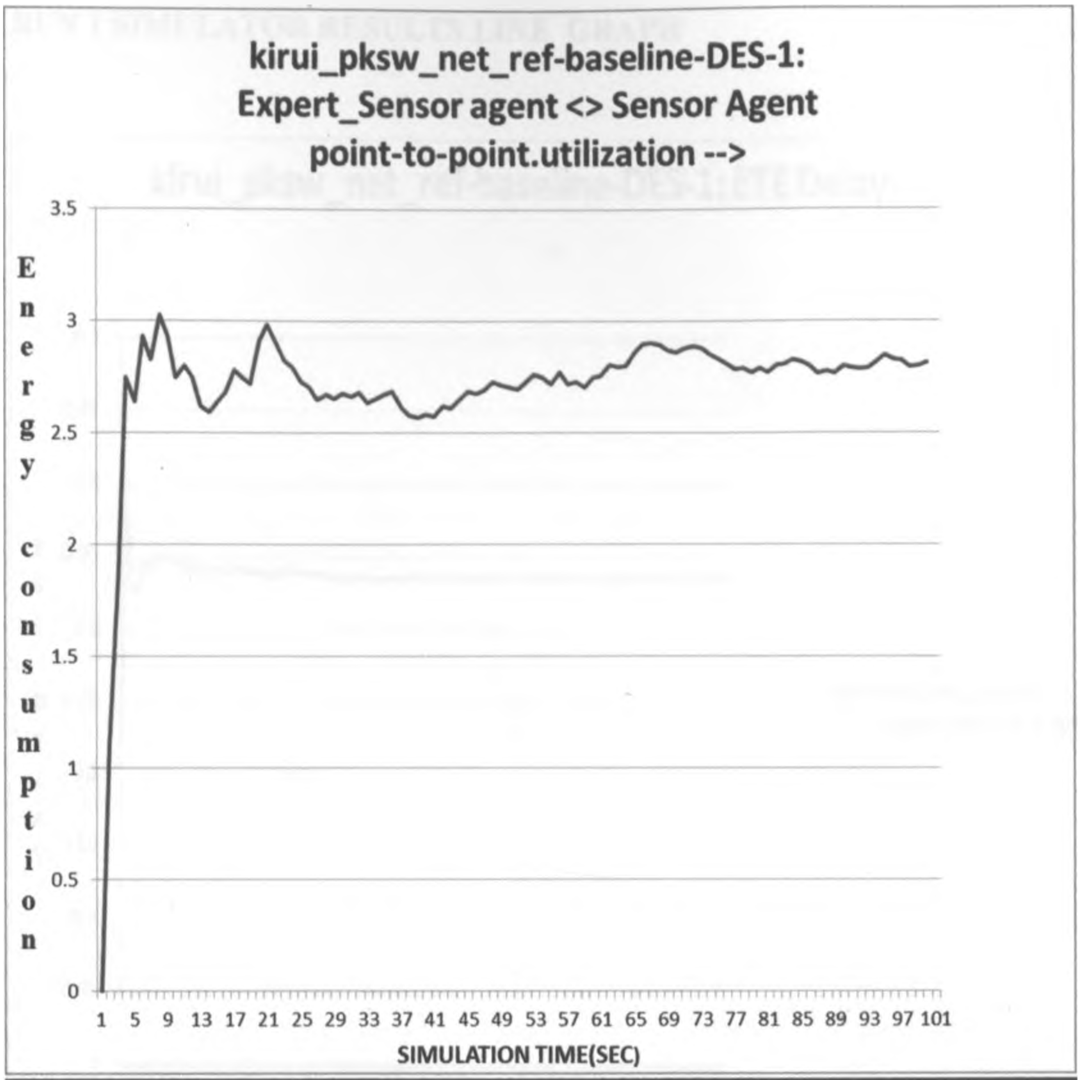


Figure 8: Point to Point Utilization

This Line graph clearly shows the discrete event OPNET simulation results for Run 1 in terms of Power Consumption performance over 100 seconds time execution.

RUN 1 SIMULATOR RESULTS LINE GRAPH

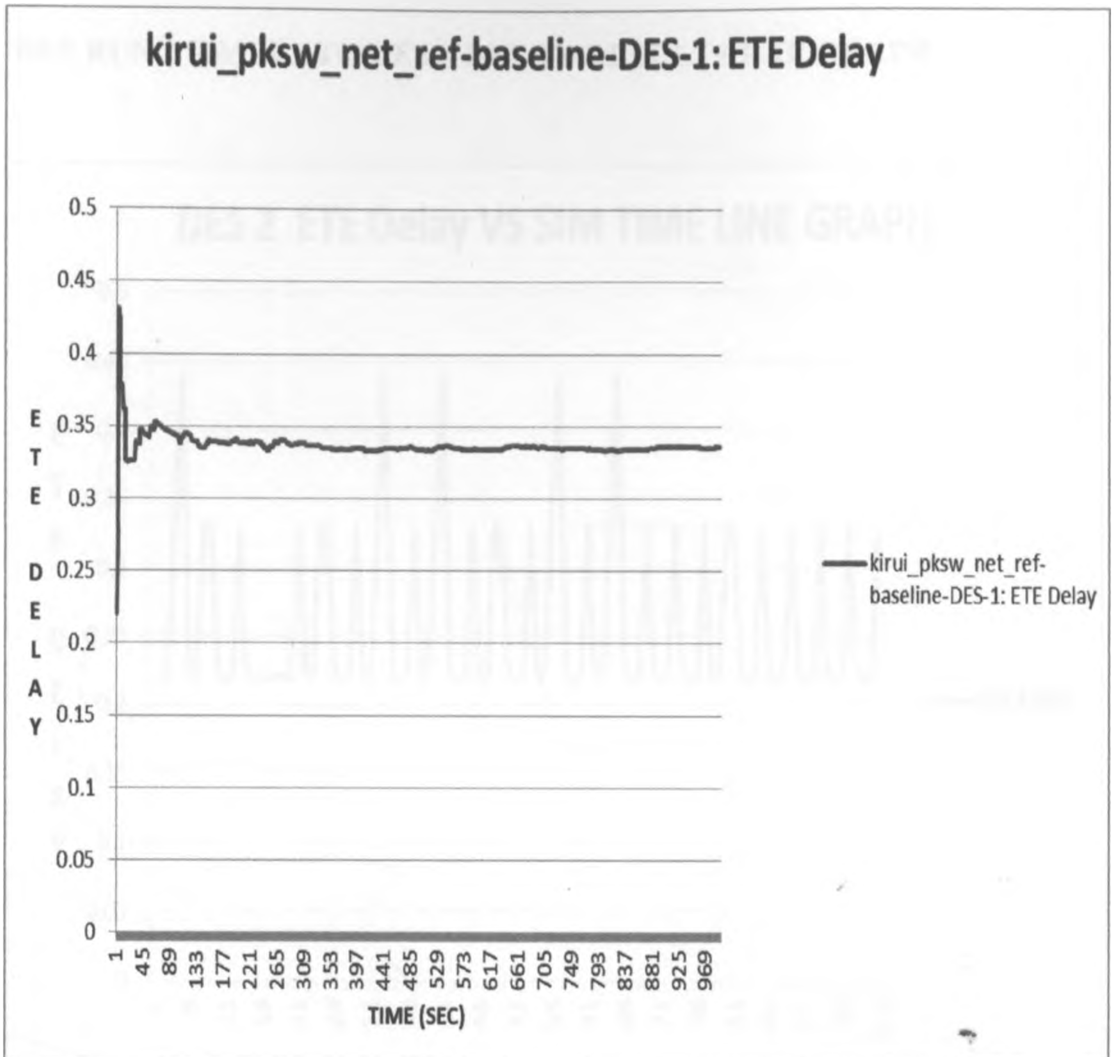


Figure 9 : End to End Delay (ETE) seconds

This Line graph clearly shows the discrete event OPNET simulation results for Run 1 in terms of end to end delay performance over 1000 seconds time execution

DES RUN 2 SIMULATOR ETE DELAY RESULTS BAR GRAPH

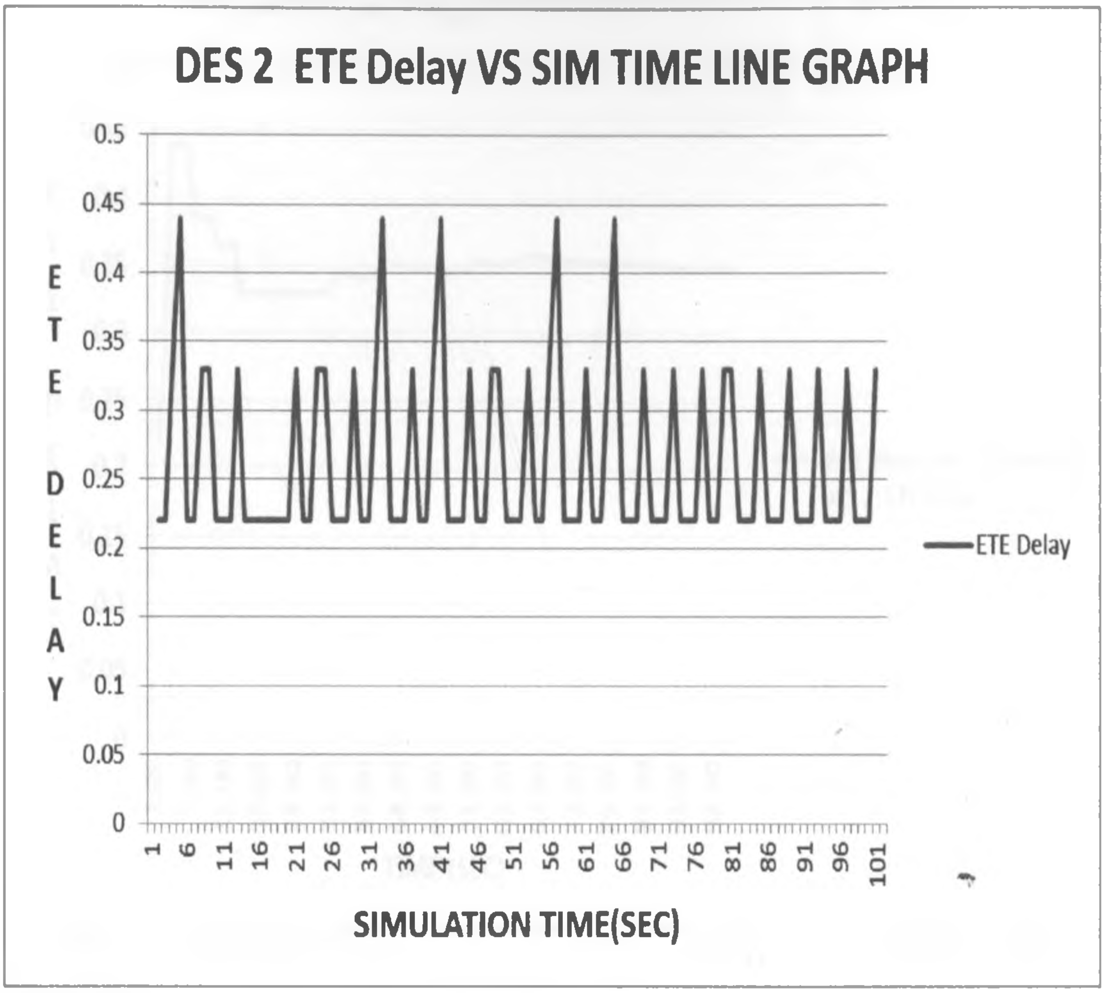


Figure 10.

This bar graph clearly shows the discrete event OPNET simulation results for Run 2 in terms of End to end delay performance over 100 seconds execution time.

DES RUN 2 SIMULATOR ETE DELAY RESULTS LINE GRAPH

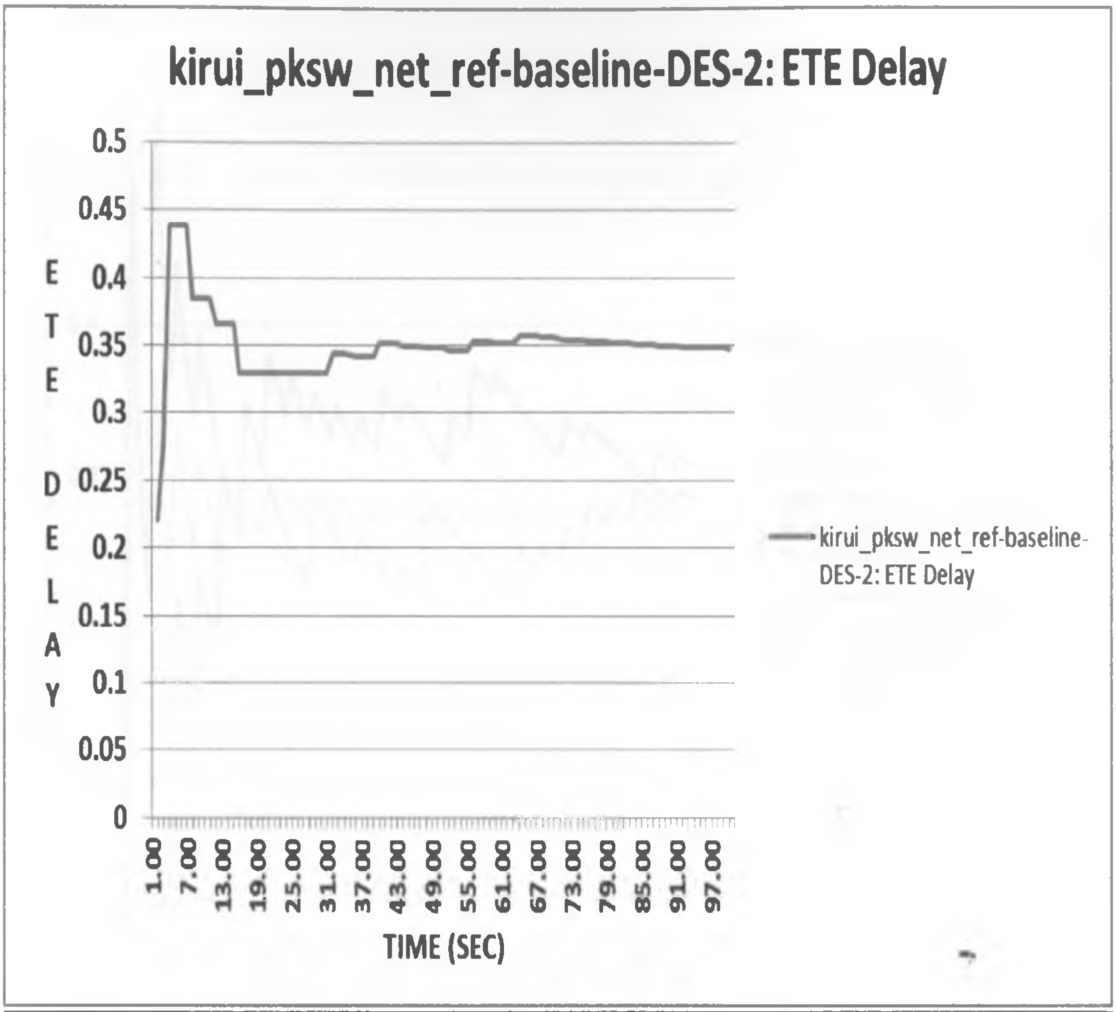


Figure 11.

This line graph clearly shows the discrete event OPNET simulation results for Run 2 in terms of End to end delay performance over 100 seconds execution time

DES RUN 2 SIMULATOR RESULTS(ENERGY CONSUMPTION VS SIM TIME)LINE GRAPH

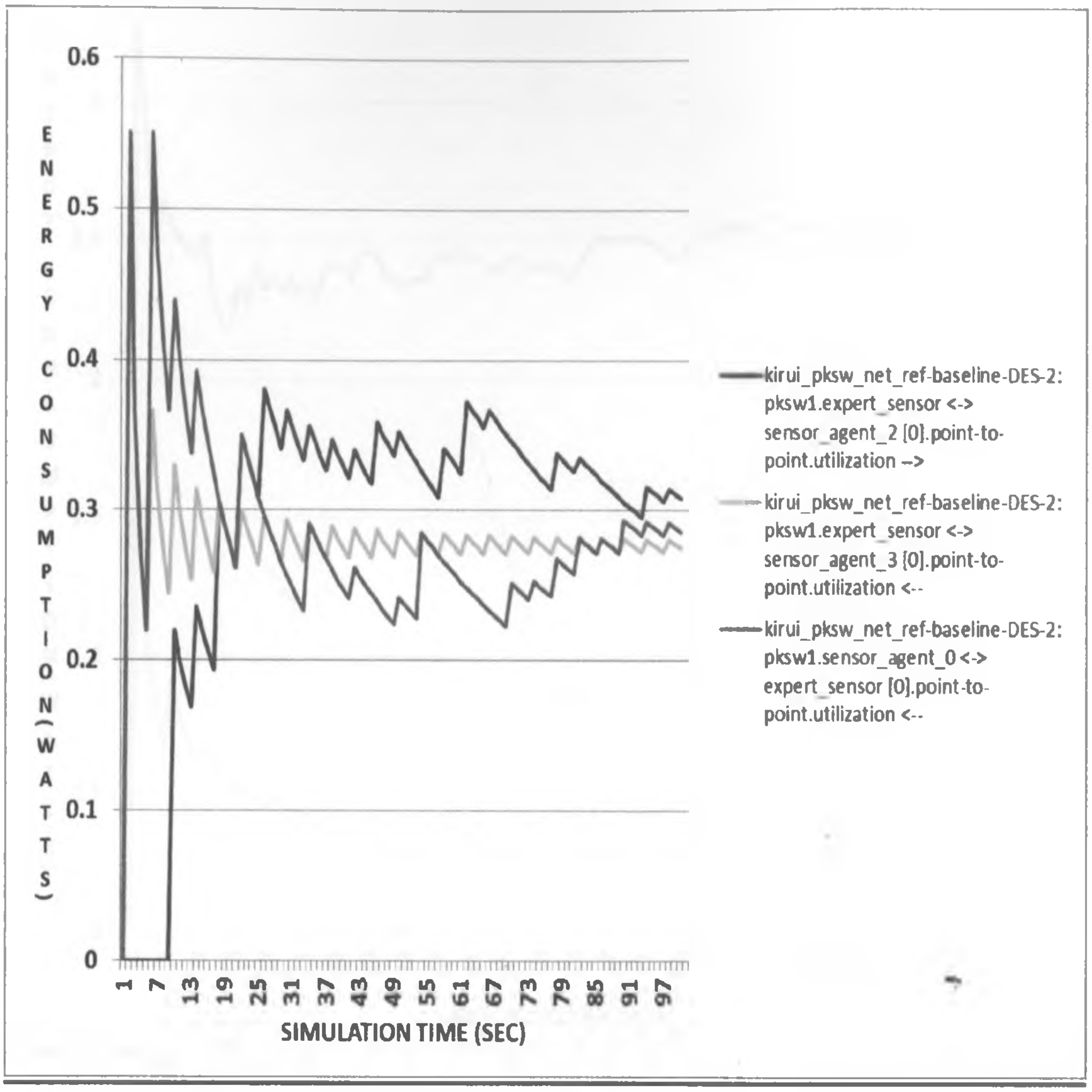


Figure 12.

This line graph clearly shows the discrete event OPNET simulation results sensor agents for Run 2 in terms of power consumption performance over 100 seconds execution time

RUN 1 & RUN 2 SIMULATOR RESULTS COMPARISON FOR (ENERGY CONSUMPTION VS SIMULATION TIME) LINE GRAPHS

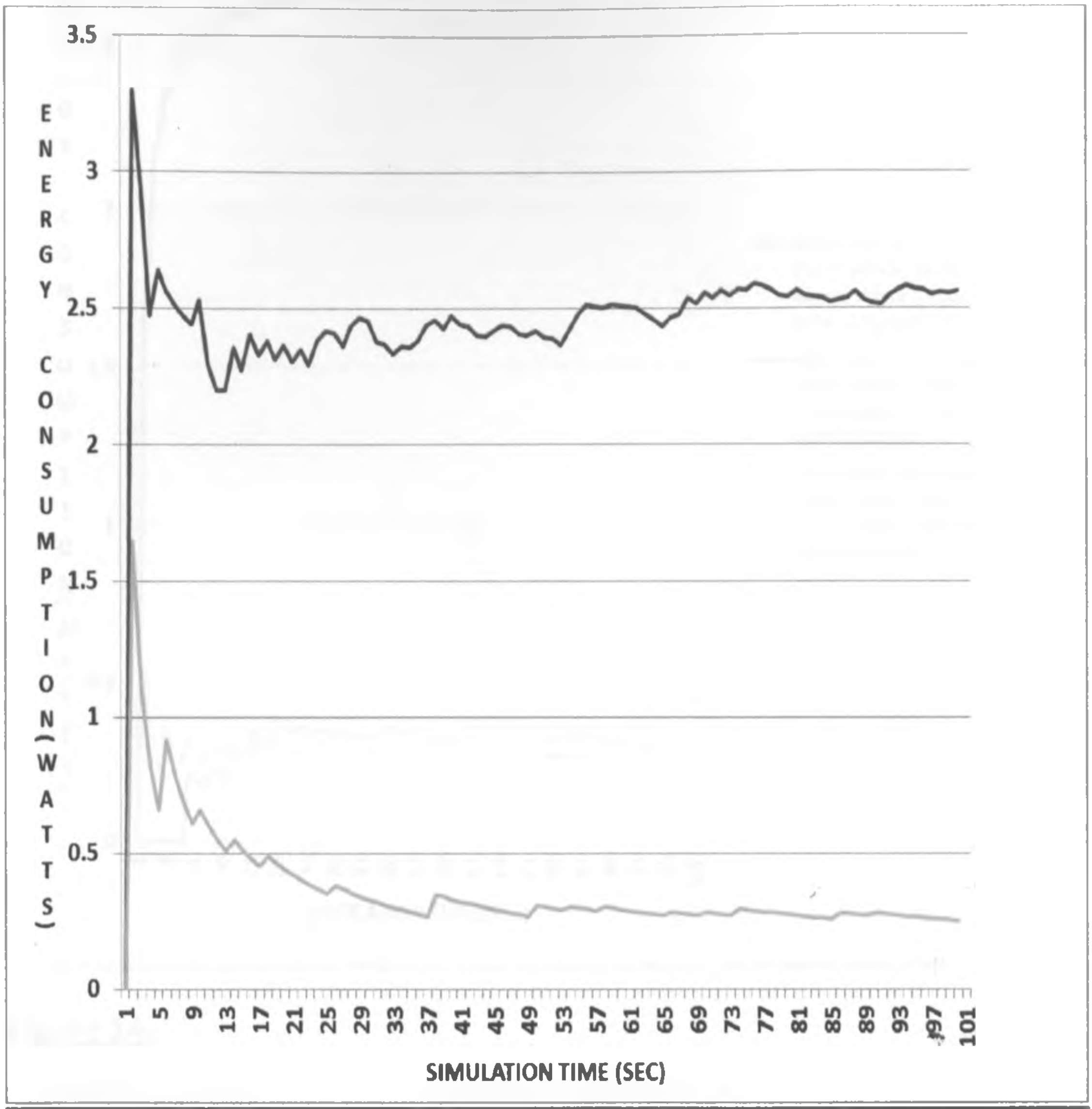


Figure 13: Power Consumption (Watts)

This line graph clearly shows the discrete event OPNET simulation results for Run 1 and Run 2 in terms energy consumption comparisons over 100 seconds execution time

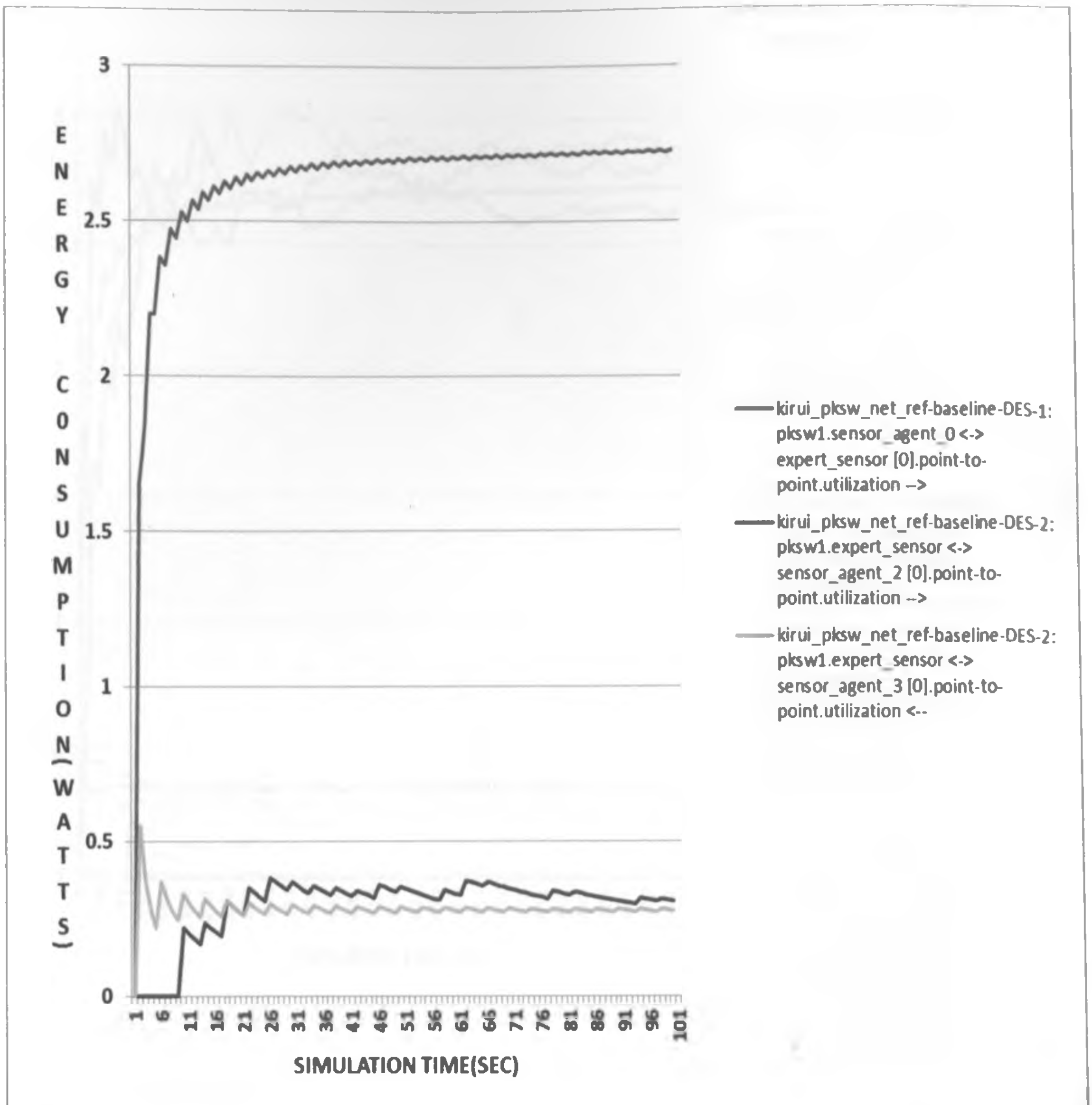


Figure 14.

This line graph clearly shows the discrete event OPNET simulation results for Run 1 and Run 2 sample sensor node agents in terms energy consumption comparisons over 100 seconds execution time

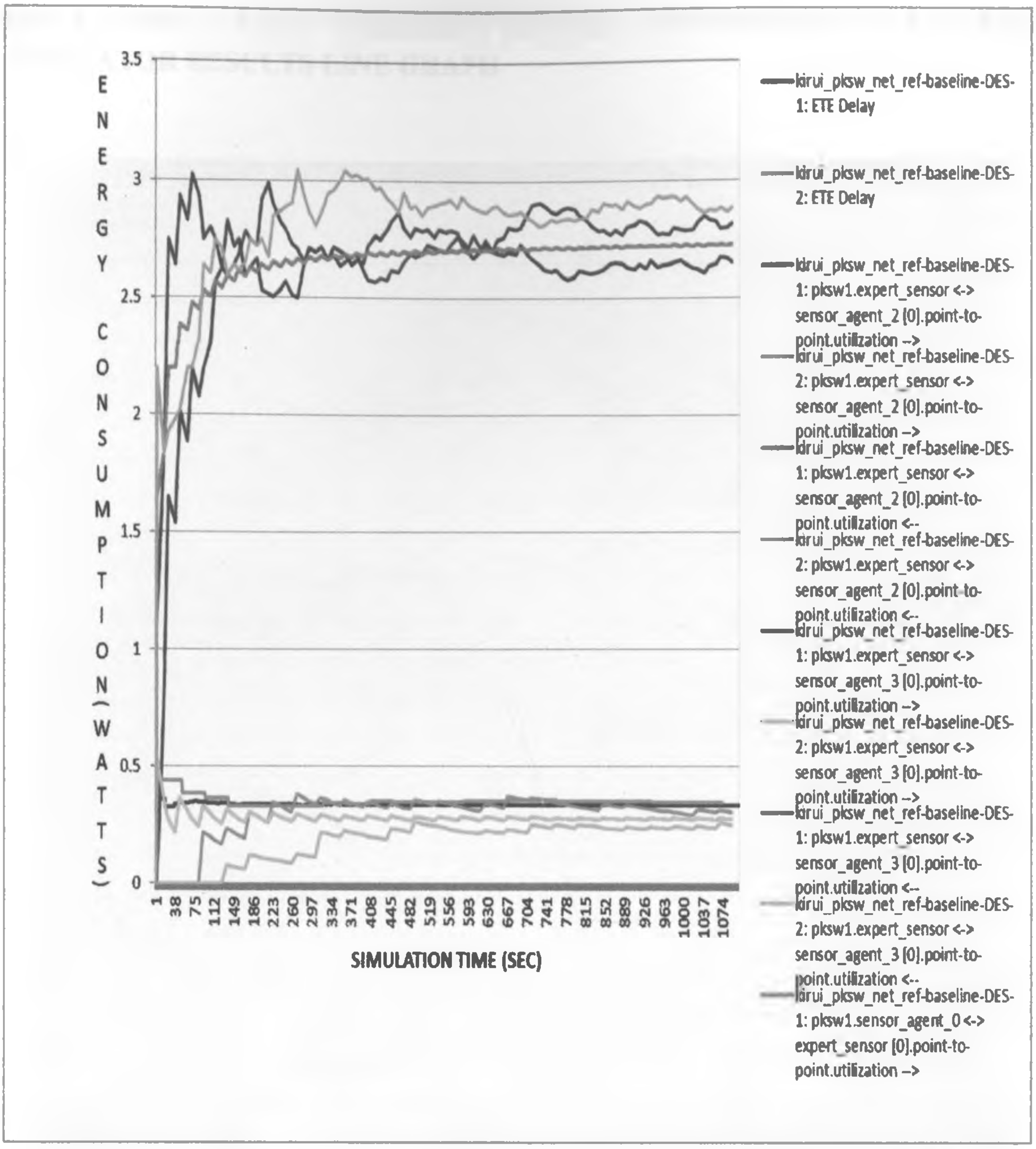


Figure 15.

This line graph clearly shows the discrete event OPNET simulation results for Run 1 and Run 2 for all sensor node agents in terms energy consumption comparisons over 1000 seconds execution time

DES 1 & DES 2 RUNS SNAPSHOT POWER CONSUMPTION VS ETE DELAY SIMULATOR RESULTS LINE GRAPH

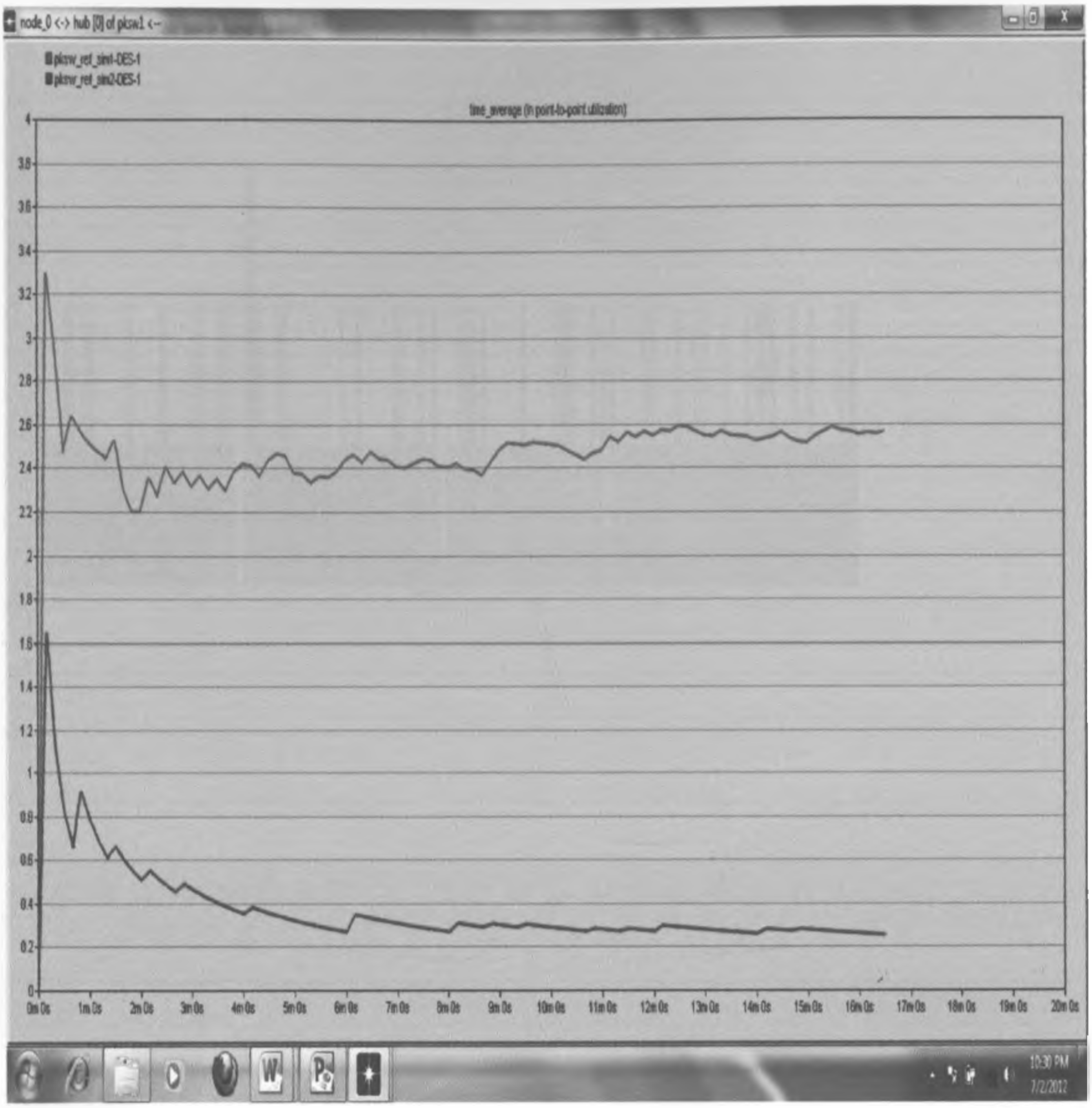


Figure 16.

DES 1 SNAPSHOT ETE DELAY SIMULATOR RESULTS BAR GRAPH

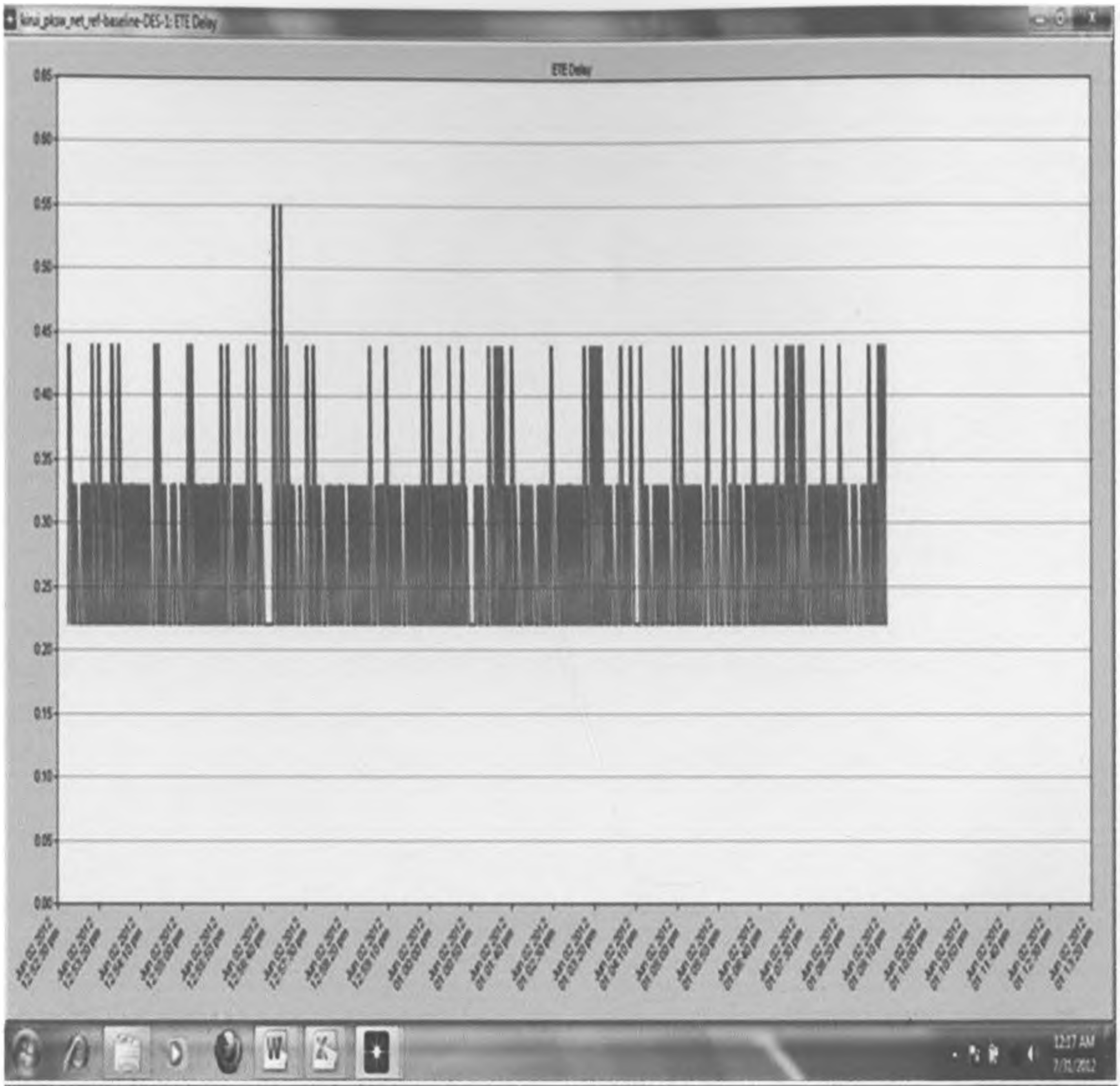


Figure 17.

DES 2 SNAPSHOT SIMULATOR ETE DELAY RESULTS BAR GRAPH

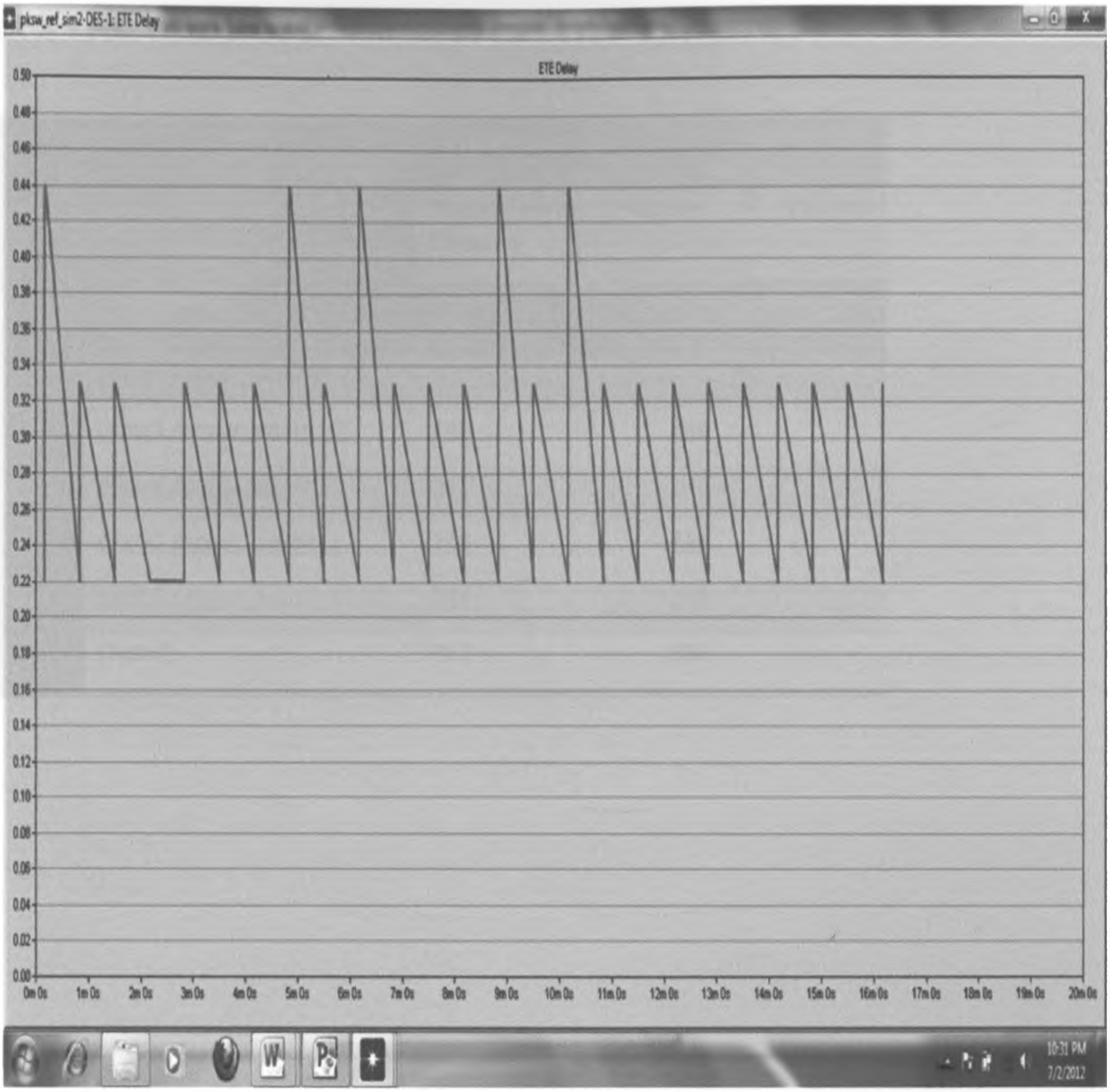


Figure 18.

SIMULATION RUNS PACKETS STATISTICS

DES RUNS 1 PROCESS MODEL PACKETS CREATED STATISTICS

Category:		Packet Statistics	
Report:		Node-centric. Number of Packets Created	
Line NO	Node Name	[Total]	pksw_format_ref
0	pksw1.expert_sensor	0	
1	pksw1.sensor_agent_0	248	248
2	pksw1.sensor_agent_1	248	248
3	pksw1.sensor_agent_2	248	248
4	pksw1.sensor_agent_3	248	248
5	[Total]	992	992

DES RUNS 1 PROCESS MODEL PACKETS DESTROYED STATISTICS

Category:	**Packet Statistics**
Report:	**Node-centric. Number of Packets Destroyed**

Line NO	Node Name	[Total]	pksw_format_ref
0	pksw1.expert_sensor	0	
1	pksw1.sensor_agent_0	262	262
2	pksw1.sensor_agent_1	233	233
3	pksw1.sensor_agent_2	256	256
4	pksw1.sensor_agent_3	241	241
5	[Total]	992	992

DES RUNS 1 SENSOR NODE PACKETS CREATED STATISTICS

Category:		pksw_format_ref		
Report:		**Number of Packets Created**		
0	pksw1.expert_sensor	0		
1	pksw1.sensor_agent_0	248		248
2	pksw1.sensor_agent_1	248		248
3	pksw1.sensor_agent_2	248		248
4	pksw1.sensor_agent_3	248		248
5	[Total]	992	992	

DES RUNS 1 SENSOR NODE PACKETS DESTROYED STATISTICS

Category:		pksw_format_ref		
Report:		Number of Packets Destroyed		
0	pksw1.expert_sensor	0		
1	pksw1.sensor_agent_0	262		262
2	pksw1.sensor_agent_1	233		233
3	pksw1.sensor_agent_2	256		256
4	pksw1.sensor_agent_3	241		241
5	[Total]	992	992	

DES RUNS 2 PROCESS MODEL PACKETS CREATED STATISTICS

Category:	**Packet Statistics**
Report:	**Node-centric. Number of Packets Created**

Line NO	Node Name	[Total]	pksw_format_ref
0	pksw1.expert_sensor	0	
1	pksw1.sensor_agent_0	25	25
2	pksw1.sensor_agent_1	25	25
3	pksw1.sensor_agent_2	25	25
4	pksw1.sensor_agent_3	25	25
5	[Total]	100	100

DES RUNS 2 PROCESS MODEL PACKETS DESTROYED STATISTICS

Category:	**Packet Statistics**
Report:	**Node-centric. Number of Packets Destroyed**

Line NO	Node Name	[Total]	pksw format ref
0	pksw1.expert_sensor	0	
1	pksw1.sensor_agent_0	26	26
2	pksw1.sensor_agent_1	23	23
3	pksw1.sensor_agent_2	28	28
4	pksw1.sensor_agent_3	23	23
5	[Total]	100	100

DES RUNS 2 SENSOR NODE PACKETS CREATED STATISTICS

Category:	**pksw_format_ref**
Report:	**Number of Packets Created**

Line NO	Node Name	[Total]	hub	proc	rcv	src	xmt
0	pksw1.expert_sensor	0					
1	pksw1.sensor_agent_0	25				25	
2	pksw1.sensor_agent_1	25				25	
3	pksw1.sensor_agent_2	25				25	
4	pksw1.sensor_agent_3	25				25	
5	[Total]	100			100		

DES RUNS 2 SENSOR NODE PACKETS DESTROYED STATISTICS

Category:	pksw_format_ref
Report:	**Number of Packets Destroyed**

ine NO	Node Name	[Total]	hub	proc	rcv	src	xmt
0	pksw1.expert_sensor	0					
1	pksw1.sensor_agent_0	26		26			
2	pksw1.sensor_agent_1	23		23			
3	pksw1.sensor_agent_2	28		28			
4	pksw1.sensor_agent_3	23		23			
5	[Total]	100	100				

Appendix 2:Source Code

OBJECT C++ FUNCTION BLOCK CODE

```
static void route_pk (void)

        {

        int dest_address;

        int energy_level;

        int MIN_SENSOR_ENERGY_LEVEL=10;

        Packet * pkptr;

/* The first line after FIN(route_pk()) has two effects: first, it retrieves the arriving packet from the appropriate input stream (whose index is determined by op_intrpt_strm()). Then, op_pk_get() uses the packet-stream index argument to return a pointer to the packet. */

        FIN (route_pk ());

        pkptr = op_pk_get (op_intrpt_strm ());


        // obtain the destination address contained in the packet.

        op_pk_nfd_get_int32 (pkptr, "dest_address", &dest_address);


//obtain energy_level of packet source node

        op_pk_nfd_get_int32 (pkptr, "energy_level", &energy_level);

        energy_level--;

        if(energy_level > MIN_SENSOR_ENERGY_LEVEL)

        {

// update energy level in the packet op_pk_nfd_set_int32(pkptr,"energy_level",

   energy_level);

//route packet to the new sensorop_pk_send (pkptr, dest_address);
```

```
}
FOUT;
}
```

SENSOR AGENT NODE C++ FUNCTION CODE

```
static void xmt (void)
        {
        Packet * pkptr;
        FIN (xmt ());
        pkptr = op_pk_get (SRC_IN_STRM);
        op_pk_nfd_set_int32 (pkptr, "dest_address",
                (int)op_dist_outcome (address_dist));
        op_pk_send (pkptr, XMT_OUT_STRM);
        FOUT;
        }
static void rcv (void)
        {
        Packet * pkptr;
        double ete_delay;
        FIN (rcv ());
        pkptr = op_pk_get (RCV_IN_STRM);
        ete_delay = op_sim_time () -
                op_pk_creation_time_get (pkptr);
        op_stat_write (ete_gsh, ete_delay);
        op_pk_destroy (pkptr);
        FOUT;
```

```
}
```

SENSOR AGENT NODE SIMPLE SOURCE C++ CODE

```
static void
ss_packet_generate (void)
    {
    Packet* pkptr;
    double  pksize;
    /** This function creates a packet based on the packet generation          **/
    /** specifications of the source model and sends it to the lower layer.   **/
    FIN (ss_packet_generate ());
    /* Generate a packet size outcome.                                         */
    pksize = (double) ceil (oms_dist_outcome (pksize_dist_ptr));
    /* Create a packet of specified format and size.          */
    if (generate_unformatted == OPC_TRUE)
        {
        /* We produce unformatted packets. Create one.       */
        pkptr = op_pk_create (pksize);
        }
    else
        {
        /* Create a packet with the specified format. */
        pkptr = op_pk_create_fmt (format_str);
        op_pk_total_size_set (pkptr, pksize);
        }
    /* Update the packet generation statistics.                   */
    op_stat_write (packets_sent_hndl, 1.0);
    op_stat_write (packets_sent_hndl, 0.0);
```

```
op_stat_write (bits_sent_hndl, (double) pksize);

op_stat_write (bits_sent_hndl, 0.0);

op_stat_write (packet_size_hndl, (double) pksize);

op_stat_write(interarrivals_hndl,next_intarr_time);

/* Send the packet via the stream to the lower layer.   */

op_pk_send (pkptr, SSC_STRM_TO_LOW);


FOUT;

}
```