Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization

Geleta Tilahun Nemera

A Thesis Submitted to the Department of Computer Science and Engineering,
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters
in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization

Geleta Tilahun Nemera

Advisor: Ketema Adere Gemeda (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters
in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June, 2023

Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled “**Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization**” is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Geleta Tilahun Nemera	_________	_________
Name of Student	Signature	Date

Recommendation

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization**" prepared under my/our guidance by **Geleta Tilahun Nemera** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in **Computer Science and Engineering**. Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Ketema Adere Gemeda (Ph.D.)	________________	______________
Major Advisor	Signature	Date
__________________	_______________	______________
Co-advisor	Signature	Date

Approval Page

I/we, the advisors of the thesis entitled "**Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization**", hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Ketema Adere (Ph.D.)	______________	______________
Major Advisor	Signature	Date
______________	______________	______________
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Geleta Tilahun Nemera**, have read and evaluated the thesis entitled "**Designing an Energy Efficient Medium Access Control for Wireless Sensor Networks Using Short Listening Time and Clusterization**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in **Computer Science and Engineering**.

______________	______________	______________
Chairperson	Signature	Date
______________	______________	______________
Internal Examiner	Signature	Date
______________	______________	______________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis are contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

______________	______________	______________
Department Head	Signature	Date
______________	______________	______________
School Dean	Signature	Date
______________	______________	______________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

For the benefits, courage, and direction I have received along this trip, I would like to offer my sincere thanks and thankfulness to the Almighty God. This thesis would not have been feasible if not for supernatural assistance and grace. I am humbled by God's illimitable knowledge and assistance.

I would also like to extend my heartfelt thanks to Ketema Adere Gemeda (Ph.D.) whose invaluable assistance and unwavering support have been instrumental in the completion of this thesis. He has been a constant source of encouragement, offering guidance, motivation, and insightful feedback that significantly enriched the quality of this research work. His expertise and dedication have truly been invaluable, and I am grateful for the opportunity to work under his mentorship.

I would also like to extend my appreciation to my friends and family members especially my father and mother who have supported me throughout this journey. Their unwavering belief in my abilities and their encouragement during challenging times have been a constant source of motivation. Their love, understanding, and patience have been essential in helping me maintain balance and focus throughout the thesis process.

Furthermore, I would like to acknowledge the academic and administrative staff at ASTU for their assistance and support during my studies. Their professionalism and dedication to fostering a conducive learning environment have been vital in my academic growth and development.

Finally, I am grateful to all the participants who took part in the research study, as their contributions were essential in everything for this thesis. Their willingness to share their time and knowledge has been deeply appreciated.

In conclusion, this thesis is the culmination of the collective efforts and support of many individuals, whose contributions have shaped and enriched this research work. I extend my sincere gratitude to each and every person who has played a role, no matter how small, in making this thesis a reality.

TABLE OF CONTENTS

LIST OF ACRONYMS AND ABBREVIATIONS

ACK - - Acknowledgement

ACM - - Association for Computing Machinery

ACW - - Adaptive contention window

B-MAC - - Barkley MAC

BS - - Base Station

CH - - Cluster Head

CTS - - Clear To Send

CSMA - - Carrier Sense Multiple Access

CW - - Contention Window

DCF - - Distributed Coordination Function

DC-MAC - - Dynamic duty-cycle Based MAC

DIFS - - DCF Inter-frame Space

DSR - - Dynamic Source Routing

EE-MAC - - Energy Efficient MAC

EES-MAC - - Energy Efficient SMAC

ESMAC - - Enhanced Sensor MAC

IEEE - - Institute of Electrical and Electronics Engineers

IP - - Internet Protocol

LEACH-C - - Low Energy Adaptive clustering Hierarchy Centralized

MAC - - Medium Access Control

MMSPEED - - Multi-Path and Multi-Speed

MS - - Mobile Sink

NAM - - Network Animator

NACK - - Negative Acknowledgement

NS2 - - Network Simulator 2

OMNET++ - - Objective Modular Network Testbed in C++

PW-MAC - - Energy efficient predictive Wake-up MAC

QOS - - Quality of Service

RI-MAC - - Receiver Initiated MAC

RTS - - Request To Send

SLT - - Short Listening Time

SMAC - - Sensor Medium Access Control

SN - - Sensor Nodes

SYNC - - Synchronization

TCL - - Tool Command Language

TCP - - Transmission Control Protocol

WSN - - Wireless Sensor Network

XMAC - - A Short Preamble MAC for WS

Z-MAC - - A Hybrid MAC Protocol for WSN

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

WSN is composed of small, low-power, and resource-constrained SNs that collaborate to monitor physical environments and transmit collected data wirelessly. Due to the limited energy resources of SNs, energy management techniques are essential to prolong the network's lifespan. Although energy saving in WSN may be carried out at several stages of the TCP/IP protocol suite, energy conservation at the MAC layer is discovered to be the most efficient one due to its capacity of directly controlling the radio. To address the energy efficiency issue in WSNs, researchers have proposed various MAC protocols, each focusing on specific aspects and trade-offs. By allowing SNs to occasionally turn off their radios and go to sleep, SMAC helps SNs decrease the main cause of energy waste known as idle listening. And ESMAC adjusts duty-cycle in accordance with changes in the amount of energy left in the nodes; they set an energy threshold and weigh the amount of energy left in each node to determine the duty-cycle. Even though a lot of research has been done, still it is not enough to save nodes energy to extend lifetime of the networks. Therefore, to ensure a long-lived network of WSNs, we are in need of a MAC protocol that is able to improve energy efficiency by maximizing sleep duration, and minimizing idle listening. This study proposes SLT and clusterization for designing energy efficient MAC protocol. In SLT, we determine the sleeping time of a node based on the residual energy of the nodes so that it does not listen for a long time when it has less energy. In clusterization, after nodes form a cluster based on their distance from each other, the amount of energy they have is measured to select cluster head. So, the simulation results indicated that the EES-MAC improves the average throughput by 5.89% and 95.28%*, over ESMAC and SMAC. The simulation findings demonstrate that EES-MAC could successfully improve network performance when used in conjunction with SLT and the clusterization adjustment method by reducing network latency, increasing network throughput, and node energy efficiency.*

Keywords: *Clustering, Energy Efficient, MAC Protocol, Short Listening Time, WSNs*

CHAPTER ONE

INTRODUCTION

1.1. Background of the Study

Large-scale industrial applications and the army were the origins of WSNs. The Sound Surveillance System, developed by the US Army in the 1950s to track Soviet submarines and detect their speech, was the first system that was comparable to the WSN. The Defense Advanced Research Project Agency began to construct Distributed Sensor Networks by examining the benefits of Sound Surveillance Systems, and later, in the early 1980s, the actual WSN was created. At this point, WSN was well-known in academia and applied science (Labs, 2004)..

WSN technologies have developed fast over time. Battery-powered WSNs have several sensors, CPUs, and radio frequency modules. Using ubiquitous sensors called sensor nodes, which collect information from their surroundings and transfer it to sink nodes, it is possible to observe ecological phenomena throughout a vast territory. Communication between SNs is based on the integration of various sensors, ranging from simple (such as humidity, pressure, and temperature) to complex (such as localization, tracking, micro-radars, and images), allowing WSNs to keep an eye on a variety of environmental factors to gather accurate information from the field (Jawad et al., 2017).

Small, battery-powered SN in WSNs are non-rechargeable after deployment and have a finite amount of energy resources. As a result, reducing energy consumption is a key consideration in the development of WSN protocols. Data routing and sensing channels both need energy. Energy-aware routing protocols have been developed by a number of authors that take this issue into account by requiring SNs to be aware of next-hop residual energy. This increases the lifespan of the network. Energy-based routing algorithms frequently employ clustering and sleep wake-up scheduling to reduce the energy consumption of SNs.

The lack of reliable data collection methods affects the overall network's lifespan (Yalçın & Erdem, 2020). The well-known method of topology control management in WSNs is known as clustering, which clusters nodes to manage and design various tasks including managing node resources in a hierarchical and dispersed way (Daas et al., 2021; Shahraki et al., 2020). It is the process of grouping nodes in a network based on predefined criteria. The CH is the cluster member's leader, which collects data, aggregates data, and transmits data to the MS.

For effective data transmission, it is critical to use appropriate routing algorithms between SNs and MS nodes (Yalçın & Erdem, 2020).

Sleep wakeup scheduling is a more effective solution to the energy consumption issue in WSNs (Wei et al., 2002). However, current MAC protocols employ idle listing, which uses almost the same amount of energy as receiving a packet. Additionally, it makes use of idle listing and overhearing, which allow SNs in a network to receive packets intended for their adjacent nodes. Wie Yie created SMAC, to increase the energy efficiency of WSNs. By allowing SNs to occasionally turn off their radios and go to sleep, SMAC helps SNs decrease the main cause of energy waste known as idle listening. SMAC is a contention-based MAC protocol with capabilities including message forwarding, periodic sleep and wake, overhearing avoidance, and collision avoidance. Carrier sensing is used by the contention-based MAC protocol to assess the state of the network and provide nodes access to channels.

The primary role of the MAC protocol in sensor networks is to allocate the wireless communication channel and the resources needed for inter-SN communication (Pang et al., 2012). Additionally, it is utilized to prevent collisions in shared-medium networks by limiting access to the medium for two or more concurrently interacting nodes. The important consideration in the configuration of WSNs is low energy usage due to the SNs' limited battery life. An extended network lifespan is the outcome of efficient energy utilization. Therefore, reducing energy consumption and enhancing network lifespan are the primary goals in the algorithm's design. Three different MAC protocol types exist. The first group is contention-based protocols, followed by scheduled-based protocols, and the third category is hybrid protocols, which combine contention-based and scheduled-based protocols.

Contention-based protocols are scalable and can perform well despite variations in the traffic or node density. However, these methods are vulnerable to substantial energy loss through collisions and idle listening. Scheduled-based protocols use less energy since they are collision-free. However, they are not very adaptable or scalable. Contention-based and contention-free protocols are both used in hybrid MAC protocols. The benefits of both types of protocols are included in these protocols. Recently, MAC protocol development in wireless networks has employed quorum systems (Shafiullah et al., 2013).

According to changes in the nodes' remaining energy, (Teshome, 2018) adjusted the SMAC duty cycle accordingly. To do this, they set an energy threshold and compared the nodes' individual remaining energies to determine the duty cycle. When it is impracticable to change the batteries for an extended period of time, the suggested EES-MAC protocol enhances the energy efficiency of WSN applications. We lengthen the sleeping period and modify the node cluster according to the nodes' residual energy and distance between them.

1.2. Motivation of the Study

One of the major concerns in WSNs is the limited energy resources of SNs, as they are typically powered by small batteries that cannot be easily recharged or replaced. This energy constraint poses significant challenges in terms of network lifetime, as the continuous operation of SNs is crucial for the success of WSN applications. To address this challenge, designing an energy-efficient MAC protocol becomes crucial.

The periodic sleep wake-up enables nodes to sleep and transmit data while they are listening to the channel at the same time introduced by (Wei et al., 2002). The energy used for idle listening, which is a major energy consumer in WSN, is reduced by periodic sleep and listening in SMAC, especially when the network traffic is low. As a result of the retransmission of dropped packets, SMAC makes it stationary and unable to respond to network changes, which might increase end-to-end delay and energy usage. (Teshome, 2018) has created a method that makes use of network size and a node's remaining energy to change the contention window size and duty cycle of the SMAC protocol in order to solve the issue with the fixed contention window size and fixed duty cycle of SMAC. Even though ESMAC lowers the duty cycle based on the amount of energy left in the nodes, nodes still wait for activity within the specified duty cycle, which affects the network's longevity. The aforementioned problems are the main driving forces behind this study's proposal to create energy-efficient medium access protocols for WSNs by utilizing SLT and clusterization to lower energy consumption and lengthen the network's lifetime.

1.3. Statement of the Problem

To tackle the energy problem in WSNs, various research directions can be pursued. These include the development of energy-efficient routing protocols, adaptive power management schemes, and novel energy harvesting techniques. Additionally, optimizing data aggregation, compression, and transmission strategies can significantly reduce energy consumption. The utilization of sleep scheduling algorithms, duty cycling, and node clustering approaches can also contribute to energy savings by intelligently activating and deactivating SNs based on the application necessities. Furthermore, the exploration of renewable energy sources, such as solar or kinetic energy harvesting, holds promise for addressing long-term energy sustainability in WSNs.

The importance of MAC scheduling at SNs cannot be overstated because it guarantees minimal energy consumption and provides QOS metrics like throughput and delay in delivering sensed data to the sink node, which reduce sensor energy consumption and data transmission delay. SMAC, which makes advantage of periodic sleep-wake scheduling, was proposed to address the aforementioned issues (Wei et al., 2002). By adjusting the duty cycle of the nodes, SMAC lowers the energy depletion of the nodes utilizing fixed periodic sleep-wake scheduling. SMAC reduces node energy consumption when the duty cycle is low, but it is anticipated that it will perform better in terms of latency and throughput when the duty cycle is high. However, because the duty cycle of the SMAC is constant, in any of the cases mentioned above, SMAC loss either saves energy or offers greater throughput and little latency.

ESMAC created a method that takes into account network size and a node's remaining energy to change the contention window size and duty cycle of the SMAC protocol in order to solve the issue with the fixed contention window size and fixed duty cycle of SMAC (Teshome, 2018). Since each node must wait for a certain amount of time during the contention window before sending or receiving SYNC packets, DATA packets, or control packets, the larger value of the contention window for a small number of nodes results in prolonged channel listening, while the smaller value of the contention window for a large number of nodes causes intense computation among SNs, which causes packet loss. Additionally, with ESMAC, the sender delivers data without verifying the application data values at predetermined intervals of time.

In this research, we have proposed the following research problems on the existing works:

- Even if nodes are not doing activities, they must complete their allotted time in the duty cycle to sleep. This exposes nodes to idle listening.
- Unbalanced consumption of energy among SNs with high and low remaining energy.

The above issues are fixed by the designed Energy Efficient MAC for WSN using SLT and Clusterization.

1.4. Research Questions

This study answered the following research questions:

RQ1: What approaches can be taken to develop a new algorithm that enhances the performance of ESMAC?

RQ2: What is the procedure for simulating EES-MAC using SLT and clusterization algorithm?

RQ3: How do we compare the performance of EES-MAC with the existing ESMAC and SMAC?

1.5. Objectives

1.5.1. General Objectives

The general objective of the research is designing an Energy Efficient MAC for WSN by setting SLT, and Clusterization.

1.5.2. Specific Objectives

The specific objectives are:

- To design the new algorithm of duty-cycle based on remaining energy and SLT for improving the performance of ESMAC.
- To design the new algorithm of clusterization based on inter-cluster distance, and remaining energy of nodes for improving the performance of ESMAC.
- To simulate the proposed algorithm using simulation tools.
- To evaluate the performance of the EES-MAC with the existing ESMAC and SMAC.

1.6. Research Methodology

The proposed research is done through the following scientific research method.

1.6.1. Literature Review

Identification of the study field is the first and most fundamental stage in narrowing down and defining the extent of the topic. To accomplish the aforementioned research goals and acquire the essential data for the study, various research resources were examined, including journal articles, conference papers, theses, and relevant books. These materials were sourced from different outlets such as IEEE, Science Direct, and Google Scholar. Generally, this phase focuses on reviewing resources on MAC protocols those done on the improvement of energy, and those that have a concept of clusterization.

1.6.2. Problem Identification

The issue that has to be addressed and solved will be articulated explicitly after studying several articles repeatedly, which aids in pinpointing the fundamental gap. We were able to identify the scarcity of energy which affects the overall network performance, quality of service, and the ability to meet application requirements.

1.6.3. Design and Develop the Algorithm

We built and developed an algorithm of EES-MAC by using SLT and Clusterization parameters that handle the issues we described by taking a close look at several alternative viewpoints on how to fix the issues from various sources. The utilization of an SLT algorithm aims to reduce the energy consumed during idle listening, which is a significant contributor to overall energy consumption in WSNs and clusterization algorithm which minimize the overall communication overhead, as cluster heads play a pivotal role in aggregating and forwarding data to the sink.

1.6.4. Simulating the Algorithm

To evaluate the performance and efficiency of the proposed algorithm, a simulation process was conducted using a combination of hardware and software tools. The software components utilized in the simulation included an OS, simulation tool, programming language, and drawing tool.

For the OS, Ubuntu 22.04.2 LTS was chosen as the platform to create a stable and reliable environment for running the simulations. The simulation tool employed was NS2 which is a widely used and well-established network simulation software. NS2 provides a comprehensive set of features and functionalities for simulating network protocols and behaviors. It offers flexibility in modeling various aspects of WSNs, making it an ideal choice for this research (Zhang et al., 2009).

The programming language used for implementing the algorithm and interacting with the NS2 simulation tool was OTCL (Object Tcl) and C++. OTCL is a scripting language specifically designed for network simulations in NS2, allowing the researcher to define network topologies, parameters, and behaviors. C++ was utilized for more complex algorithm implementation and performance optimization.

Additionally, Draw.io, a diagramming tool, was utilized to create architectural diagrams and figures related to the study. This tool facilitated the visual representation of the proposed algorithm, network topologies, and other relevant illustrations, aiding in the comprehension and communication of the research findings (Engelmann et al., 2022).

In terms of hardware, the simulation process was conducted on a system with the following specifications, which have sufficient computing power and storage capacity to execute the simulations effectively.

- ✓ RAM: 8.00 GB
- ✓ System Type:x64
- ✓ Hard Disk: 1TB

The simulation process allowed us to create a controlled environment in which the proposed algorithm could be tested and evaluated. By running the algorithm within the simulated network environment, the performance metrics such as throughput, delay, and Energy Consumption could be measured, analyzed, and compared against the existing system.

1.6.5. Result Evaluation and Analysis

The proposed solution, referred to as Designing an Energy Efficient Sensor MAC, has been compared to ESMAC and SMAC. Chapter four of the study focuses on simulating and presenting the enhanced outcomes achieved through simulation. Three network performance metrics, namely energy consumption, throughput, and delay are employed to simulate the suggested technique.

1.7. Significance of the Study

WSNs are employed in a variety of applications, including surveillance, the military, agriculture, and the detection of forest fires. Routing and medium access control protocols should manage communication among SNs since they are compact and capable of forwarding sense data in multi-hop settings. Each WSN application has its own specifications. Since the majority of essential event reporting should be done when a fire is spotted in the forest, an application like forest fire detection uses less energy. When there is no fire, the sensors in these regions should conserve energy, but when a fire is detected, they must send data as quickly as necessary. Health monitoring is one use of wireless sensors that calls for better throughput, minimal latency, and energy economy. In order to maximize throughput while lowering latency, we have improved the ESMAC protocol by changing SLT and CWs dependent on network size. SNs' duty cycle is dynamically adjusted according on the amount of energy they still have, which lowers their energy usage.

1.8. Scope and Limitations of the Study

1.8.1. Scope

The scope of this study is delimited to WSNs to efficiently use scarce resources of wireless SNs through developing MAC layer protocol that considers SLT, clusterization, and cluster head selection of nodes in a network. This study exploits the weakness of ESMAC scheduling schemes and designs a proper scheduling algorithm to overcome this weakness.

1.8.2. Limitations of the Study

Due to the requirement of additional study and scope constraints, EES-MAC security issues, and real deployment of the network were not included in this thesis.

1.9. Application of the Study

This study contributes to ESMAC and SMAC in WSN using short listening time and clusterization. It addresses the energy problem in MAC protocol. In this work, SLT is used to minimize energy consumption and select a cluster head based on residual energy and inter-cluster distance. This work is crucial for numerous applications, including the military's need to identify enemy movement and intrusions, the environment's need to detect forest fires and prevent natural disasters, to monitor air quality, water quality, temperature, humidity, and pollution levels; in industries, the sensors could sense any leakage and raise alert before any disasters event like fire; in agriculture, the sensor enables continuous monitoring, data collection, and precise control of irrigation and fertilization systems, etc. and researchers can utilize it as a starting point for better optimization.

1.10. Organization of the Study

The following five chapters make up this thesis. Each chapter is organized on a distinct subject. Introduction, statement of the problem, motivation, objectives, methods, scope, limitation, and application of the thesis work are all included in chapter one. Literature reviews and related works are briefly covered in chapter two. The part provides the reader with a broad hint about the study field. It addressed the issue that the thesis is intended to fix. The third chapter describes the proposed EES-MAC, including its' general working principle and architecture. Simulation analysis and outcome evaluation are covered in chapter 4. The last chapter summarizes the findings, offers suggestions for further research, and conclusion.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

The literature review and related work section of the thesis provide an in-depth analysis of existing research and developments related to energy-efficient MAC protocols in WSNs. It seeks to determine the present level of knowledge in the area, spot any gaps in or restrictions on current methods, and emphasize the importance of the proposed study.

The section begins by introducing the fundamental concepts, applications, architecture, characteristics, design challenges, and routing protocols of WSNs. It provides an overview of the key objectives and challenges in designing energy-efficient MAC protocols, such as optimizing energy consumption, extending network lifetime, ensuring reliable communication, and accommodating the dynamic nature of WSNs.

Next, the section presents a comprehensive review of relevant literature and research papers. It includes a variety of MAC protocols that have been put out and researched in relation to WSN energy efficiency. This includes both classical protocols and more recent advancements in the field. The review encompasses various approaches such as contention-based protocols, duty-cycling techniques, and adaptive schemes.

The evaluation highlights each protocol's essential components, performance traits, and energy-saving features while critically analyzing its advantages and disadvantages. It analyzes the benefits and drawbacks of different protocols in terms of throughput, network latency, energy efficiency, and overall system performance.

2.1. Literature Review

2.1.1. Overview of WSN

The WSNs are a part of a system made up of a sizable number of small SNs that are inexpensive, basic in design, and densely distributed to work together closely to do some challenging tasks. In contrast to the conventional network, the WSNs have the following features: data-centric routing mode, limited resource supplement, large-scale deployment, and many-to-one traffic flow pattern (Lin et al., 2020). WSNs have a wide range of applications because to their tiny size and affordable node characteristics. Particularly, in recent years, these sectors have seen their applications: environmental detection, military surveillance, healthcare, tracking endangered species, etc. (Dias et al., 2019).

However, since the SNs are often battery-powered and must operate in large numbers in difficult or even unexplored environments, it is unfeasible or impractical to resupply them after deployment (He et al., 2018; Tosi et al., 2019). The fact that batteries have a very finite amount of endurance just makes things worse. The network will be divided into several portions once the percentage of SNs that lack energy exceeds a particular level. The inability of data to reach the server, which is often positioned distant from the network region, is caused by the network partition, which also causes the network topology to become disconnected. As a result, the WSNs are unable to gather sufficient accurate data for the application, which causes the network lifespan to end prematurely.

2.1.2. Wireless Sensor Node Architecture

Wireless sensor node architecture consists of subsystems which are described in Figure. 2.1:

Sensing subsystem: The sensing subsystem includes one or more physical sensors as well as one or more analog-to-digital converters and a multiplexing technique for sharing them.

Processing subsystem: The processor subsystem combines all the other subsystems plus a few more peripherals. Its main job is to process or carry out commands linked to sensing, communication, and self-organization. A processor chip, an active memory for momentarily storing sensory data, an internal clock, and a nonvolatile memory (typically an internal flash memory) for storing program instructions make up this device.

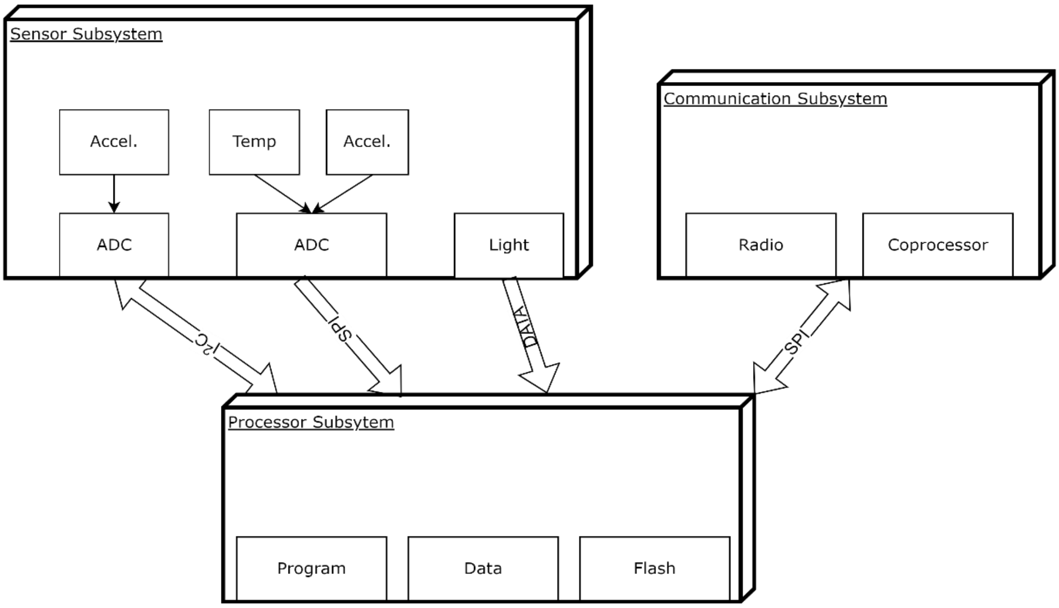


Figure 2.1 WS Node Architecture (Jondhale et al., 2022)

Communication subsystem: The efficacy of the network as a whole depends on the speed and efficiency of data exchange between the subsystems of a wireless sensor node. System buses are nevertheless constrained by the node's realistic size. In light of this, WSNs are not immediately applicable to the communication scenario on other types of networks.

2.1.3. WSN Architecture

The large-scale deployment of a wireless sensor node is to monitor a certain area. They will use routing techniques to send the data (event) they are sensing to the sink. According to the routing and other algorithms that are installed on SNs, SNs sense, record, interpret, and transmit the detected data to the sink node (Akyildiz et al., 2002). Sink nodes gather information from deployed SNs and process it to fulfill their intended purpose. The user will receive the processed data via the internet. A gateway that connects SNs to the internet can be a sink node. The sensor communicates sensed data either in single-hop (sending data directly from the source node to the sink) or multi-hop (routing source node data through intermediate nodes) scenarios according to application requirements and defined routing algorithms. i.e., while applications aimed at energy conservation route source data through intermediary nodes, applications that require little transmission delay for sensed data may transfer data directly to the sink. However, some applications require both of the application requirements. Both time-critical and non-time-critical data coexist in this application (Datta et al., 2013).

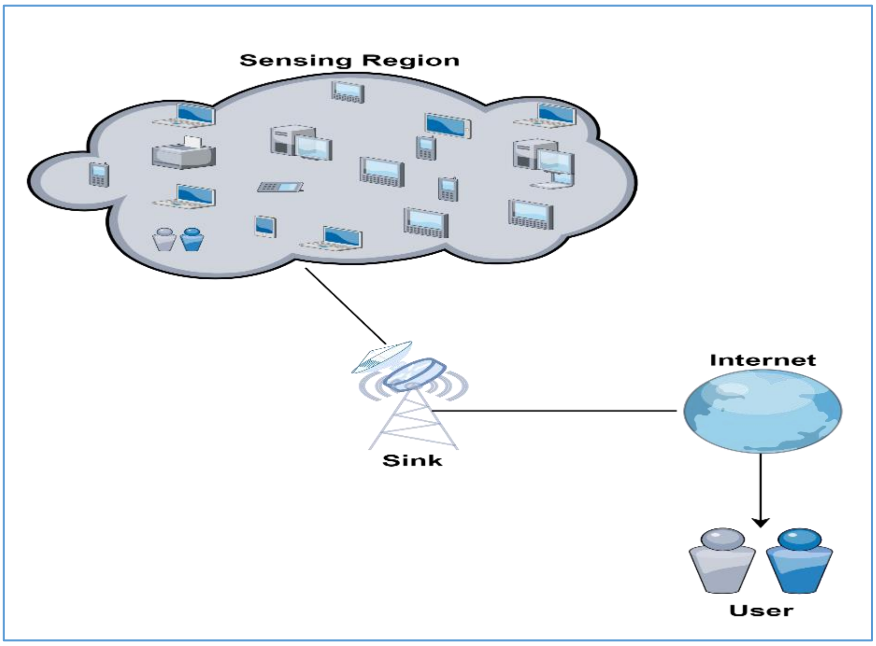


Figure 2.2 WSN architecture (Gherardi & Aquiloni, 2011; Iyekekpolo et al., 2018)

2.1.4. Characteristics of WSN

WSNs have the following unique characteristics (Buckner et al., 2016):

Self-Organization: WSNs possess the ability to self-organize, allowing nodes to autonomously establish network topologies and adapt to changing conditions. Self-organization enables efficient deployment and management of the network, as well as fault tolerance. Nodes can dynamically form clusters, elect cluster heads, and establish routing paths, adapting to environmental changes and node failures.

Scalability: WSNs can have tens of thousands or more SNs, ranging from a few to many. To handle the needed number of nodes and meet the specified coverage area, the network design should be scalable. The network's capacity to scale assures that it can manage the rising needs of data collecting and processing without compromising performance.

Heterogeneity: WSNs may exhibit heterogeneity in terms of node capabilities, such as different sensing modalities, processing power, or communication ranges. Heterogeneity enables task specialization, efficient resource allocation, and data gathering from diverse sources. Protocols and algorithms need to consider the heterogeneity of nodes to ensure optimal utilization of resources and effective network operation.

Limited Energy Resources: Small, battery-operated devices with limited energy resources are typical of SNs in WSNs. Energy efficiency becomes crucial since batteries cannot be recharged or changed after being deployed. To increase the network's lifespan, energy-saving strategies including duty cycling, data aggregation, and sleep scheduling are used.

Application-specific: A sensor network is often created and put into operation for a particular application. With each application, a sensor network's design specifications alter. Data perceived by SNs typically flow from several source SNs to a sink in sensor network applications, demonstrating a many-to-one traffic pattern.

Ad hoc Deployment: WSNs are often deployed in ad hoc or unplanned environments, such as remote areas or disaster-stricken regions. The network should be capable of self-configuring and adapting to the deployment scenario without relying on pre-existing infrastructure. This flexibility in deployment enables rapid and cost-effective deployment in diverse settings.

Fault Tolerance: Due to the large number of SNs and the challenging deployment environments, WSNs are susceptible to node failures and communication disruptions.

Ensuring fault tolerance is crucial for maintaining network connectivity and data delivery. Redundancy mechanisms, distributed algorithms, and fault detection and recovery techniques are employed to achieve fault-tolerant operation in WSNs.

Changes in topology: are frequent in WSNs because of node failures, damage, additions, energy depletion, and channel fading.

Dense sensor node deployment: SNs are typically placed close together to perceive their surroundings or the area where they are placed.

Battery-powered: Wireless sensor nodes must be powered by batteries and are frequently placed in hostile environments. They have extremely limited energy, computing, and storage capabilities due to severe restrictions.

Resource Constraints: SNs in WSNs often face resource constraints, including limited processing power, memory, and communication bandwidth. These constraints necessitate the development of resource-aware protocols and algorithms that optimize resource utilization. Techniques such as data aggregation, distributed processing, and adaptive routing are employed to mitigate resource limitations.

Data redundancy: Because SNs are deployed densely, WSNs' sensing ranges are closely correlated with one another.

Data-Centric Communication: Data collection and transmission from sensing nodes to a centralized base station or sink node are the main focuses of WSNs. Data aggregation, data fusion, and routing are just a few of the data-centric processes that the communication protocols and algorithms are built to effectively manage. Compression and in-network processing are two methods that assist minimize the quantity of data delivered, saving bandwidth and energy.

2.1.5. Application of WSNs

Wireless sensor network has various application areas. The basic application areas according to (Obaidat & Misra, 2014) are:

Military Applications: WSN can be utilized in military command, control, computing, communications, intelligence, surveillance, and other applications.

Forest fire detection: Users can be alerted to the specific cause of a fire before it gets out of hand by dispersing SNs around a forest at random and in dense concentration.

Event Monitoring: They may be used for local actuator control, continuous sensing, event detection, event ID, and position sensing.

Drug administration: The only reason hospitals administer drugs is to treat patients, and the SNs on medications make it less likely that the wrong medication would be obtained and given to them.

The home applications: Smart SNs and actuators are becoming more common in modern appliances including vacuum cleaners, microwaves, refrigerators, and VCRs. Through the Internet or satellite, these SNs inside of home appliances may communicate with one another and the outside network (Su et al., 2004). They make it simpler for end customers to locally and remotely operate household appliances.

Monitoring Environmental conditions: Wireless sensor nodes may be used to keep an eye on a variety of environmental variables. A wireless sensor network may track a variety of parameters, including temperature, humidity, vehicle movement, lightning conditions, pressure, and noise levels. It also covers environmental sensor network applications, such as monitoring weather conditions that influence crops and tracking the movements of birds, small animals, and insects (Islam et al., 2019; Sinha & Chandrakasan, 2001)

Health Monitoring: In hospitals, integrated patient monitoring, diagnostics, the dispensing of medications, the observation of insect or other small animal behavior, and other health-related uses of sensor networks are available: tracking and keeping an eye on medical staff and patients within a hospital. (Alemdar & Ibnkahla, 2007).

2.1.6. Wireless Sensor Nodes OSs

SNs in a WSN are endowed with computational and memory-intensive limited resources. A key goal is to maximize the SNs' life because it is impossible to replace them often (Abrach et al., 2003). OS designed for WSNs differs from typical OS design due to the added issues these properties of WSNs provide. The OS's purpose is to control and coordinate how users are allocated resources. An OS multiplexes system resources in both time and space. With time multiplexing, several packet transmission jobs use the resource in turn. The resource is accessed simultaneously, incase using of space multiplexing (Sinha & Chandrakasan, 2001).

Major Concerns in WSNs OS Design

According to authors (Farooq & Kunz, 2011), the following are the main concerns with WSN OS design:

Architecture: There are several types of wireless sensor architecture, including monolithic, microkernel, virtual machine, and layered. In a monolithic design, each of the OS's services is built separately and has an interface for the others. Monolithic design has minimal module interaction costs. Monolithic architecture has the disadvantage that it is hard to comprehend, adapt, and maintain. A microkernel is the design choice for many embedded OSs because of the small kernel size and the few context changes in a typical WSN application. Virtual machine architecture's main principle is to export virtual machines to user programs, which improves portability but lowering system performance. While diminishing flexibility from an OS design standpoint, the layered architecture-based OS design results in manageability, simplicity, and dependability.

Programming model: The event-driven and multithreaded programming models are those offered by common WSN OSs. Although multithreading is the application development model, it is not seen to be appropriate for devices with limited resources, such as SNs, because it is resource-intensive in its real sense. Even though it is not considered to be convenient for developers of ordinary applications, event-driven programming is deemed to be more advantageous for computing devices with limited resources. Academics have focused their efforts on developing an easy-to-use multithreading programming paradigm for wireless sensor OSs.

Scheduling: A scheduler's major goals are to reduce delay, increase throughput, and make fair use of resources. A proper scheduling technique for WSNs must be determined based on the application. WSNs are used in both real-time and non-real-time applications. Because of this, the WSN OS needs scheduling algorithms that can meet the demands of the applications. An effective scheduling strategy should also save energy and memory.

Protection and management of memory: Memory management refers to a technique for allocating and freeing memory for different processes and threads. The terms "memory management techniques" refer to both static and dynamic memory management. Static memory management is a simple and effective approach for working with memory resources that are constrained. However, it results in stiff systems since there is no run-time memory allocation. Because memory may be allocated and released during usage, dynamic memory management offers systems with greater flexibility.

Support for Communication Protocols: In a WSN, SNs communicate with other network nodes in a distributed manner. An API is provided by each OS for WSNs, allowing application applications to connect.

Resource sharing: When several applications are running at once, the OS must distribute resources and enable resource sharing. Now that the majority of WSN OSs support multithreading, a resource sharing mechanism is required.

2.1.7. OSI Layer Architecture of WSN

WSN uses the OSI concept, but there are five fundamental levels in WSNs. The application layer, transport layer, network layer, data-link layer, and physical layer are among these layers. The application layer provides software enabling various applications to send requests to get specific information and is in charge of managing traffic. Reliability and congestion avoidance are provided by the Transport Layer employing methods like ACK, NACK, and sequence number for loss detection. Techniques for recovering lost packets include end-to-end or hop-by-hop loss recovery. End-to-end dependability is less energy efficient than providing hop-by-hop reliability. Since TCP offers end-to-end reliability, therefore it is not appropriate for WSNs. UDP uses a smaller amount of memory and has less overhead (Akhmetshina et al., 2003).

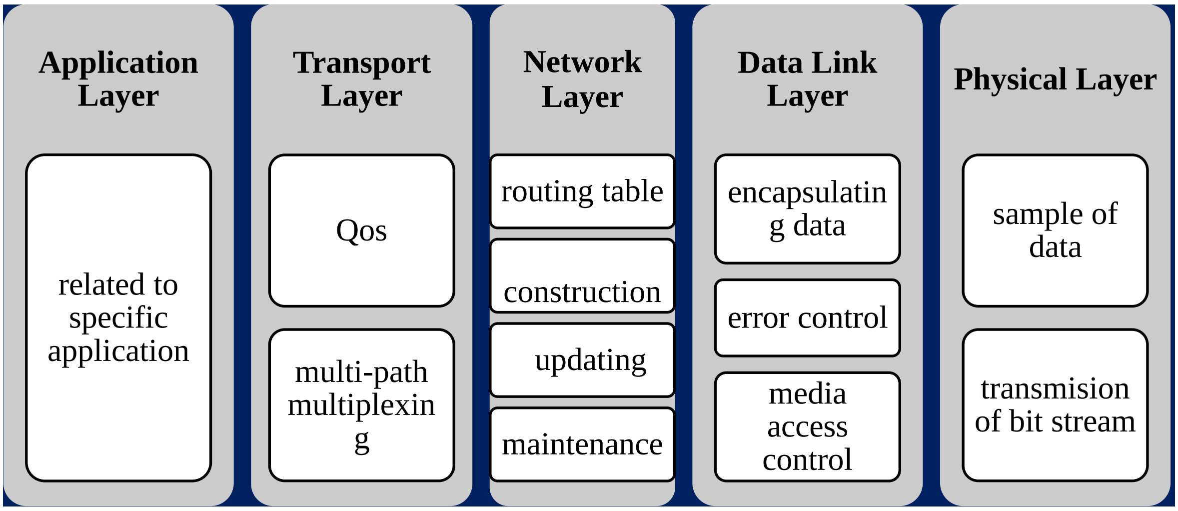


Figure 2.3 WSN OSI layer (Lin et al., 2020)

Bit streams from the data link layer must be transformed into signals that can be sent over the communication channel by the physical layer. It must deal with a number of associated challenges for this purpose, including signal modulation and detection, carrier frequency creation, data encryption, and the selection of the transmission medium and frequency. The design of the underlying hardware as well as numerous electrical and mechanical connections must also be addressed.

Network Layer is in charge of sending updates from sink (base station) to SNs in the field as well as relaying observed data to the sink node. Data fusion and aggregation are also carried out on the network layer in addition to routing. The deployment of several wireless sensors in the region to be monitored increases the likelihood that SNs may detect duplicate data (Heidemann et al., 2001). Data aggregation is used to stop redundant data from being received from nearby nodes or from being noticed by the node itself. "Data fusion" refers to additional processing done on aggregated data. Data may undergo further processing in order to get more accurate findings. The term upper layer (Network, Transport, and Application) describes layers that are above the data connection layer. Since the data link layer is the main subject of our attention, Figure 3 lists its sub-layers.

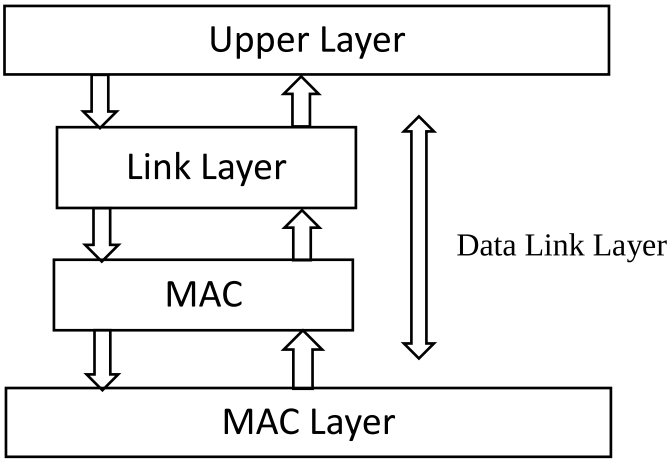


Figure 2.4 Sub-layers of data link layer (Sohraby & Al., 2007)

The data connection layer is responsible for data stream multiplexing, data frame detection, MAC, and error control. In general, the data link layer is in charge of two tasks: creating the network's framework and making sure that communication resources are distributed across SNs in a fair and efficient manner. When building MAC layer protocols for WSN, the majority of these protocols prioritize on-time delivery and place power conservation as a secondary consideration. The MAC layer protocol, which saves energy well, is unable to meet the quality of service requirements of lower delay and better throughput. Battery replacement or recharging is impossible due to the size and location of WSN. Therefore, energy use is a major issue in WSN architecture. The network designer can alter a number of physical layer variables. They are hop distance, transmission power, and modulation type.

2.1.8. WSN Protocols Design Challenges

(Fong, 2017) lists the following as the particular design issues for WSNs:

Cost of Production: Only inexpensively made sensors would be able to compete with more traditional information collecting approaches since certain deployment models consider sensors as disposable equipment. Ideally, the desired objective price for an SN should be extremely low (Sahoo & Nayak, 2019).

Culpability Tolerance: SNs are widely employed in dangerous areas and are susceptible. The nodes' failure is supposedly brought on by hardware problems, physical impairments, or overusing their energy supplies. The node failure rates are obviously far larger than those generally taken into account in reinforced or infrastructure-built WNS. The protocols used in an SN should be extremely strong in managing a proportionately high number of node failures while maintaining and retaining the entire functionality of the network system. They should also be great at quickly identifying these node failures (Hoblos et al., 2000).

Hardware Limitations: Every SN must have a power supply component, a transmission component, a processing component, and a component for information sensing. A localization system or other devices, such as a variety of internal sensors that provide location-aware routing may eventually be added to the nodes. However, every such function has a premium cost and increases the node's physical size and power use (Silva et al., 2004).

Scalability: SNs can range in size from a few to several hundreds of nodes. The deployment density can also be suitably changed. In the process of designing, deploying, and using WSNs to collect high-resolution data, the node density may rise to the point where a node has a lot of neighbors within their transmission range (Shih et al., 2001).

The Consumption of Power: As was already said, inadequate power supply were the main cause of the issues WNSs faced. The size of the battery as a power source is constrained by the size of the nodes. As a result, energy efficiency must be carefully considered while designing hardware and software. Data compression, for instance, could use less energy when being sent via radio, but it uses more energy when being processed, calculated, or filtered. Furthermore, the energy strategy is application-specific, since in certain cases it may be beneficial to turn off a subset of nodes in order to save and preserve energy, while in other cases all nodes must operate at once (Rabaey et al., 2000).

The Media of Transmission: The most frequent method of implementing communication and interaction among the nodes is radio communication utilizing the standard ISM bands. Other sensor networks, however, rely on optical or infrared communication, with infrared having the advantage of being reliable and nearly interference-free.

Topology of the Sensor Network: Although WSNs have made progress in a number of areas, these networks still have resource constraints in terms of their ability to store, process, and communicate energy. The huge number of algorithms, processes, and protocols that have been created to save energy and, as a consequence, encompass the development of the network, show that the energy resource is the most significant of all the aforementioned limits. Topology maintenance is reportedly one of the most crucial issues that might assist lower the rates of energy consumption in WSNs (Nwankwo & Ukhurebor, 2019).

2.1.9. Routing Protocols in WSNs

According to (Rastogi, 2017), the network structure model, communication model, topology, and dependability of WSN routing protocols define them. The network structure model includes hierarchical routing techniques.

Hierarchical Routing Protocols

These protocols are referred to as "Cluster Based Routing Protocols." We divide a big network into clusters, or smaller groupings, using this strategy. The SNs in a cluster whose energy is greater than the energy of the other SNs are chosen to be the Cluster Head. Direct transmission provides data directly to the sink as opposed to multi-hop transmission, which sends data via a number of intermediary nodes between the source node and Sink. Because there are many packets moving through the networks as a result and because each packet has a great distance to go, their energy is quickly used up. This method lowers power usage and message transfer to the sink by applying data fusion and aggregation during direct transmission. The five hierarchical routing protocols covered in the (Rastogi, 2017) article are as follows:

- Low Energy Adaptive Clustering Hierarchy
- Low Energy Adaptive clustering Hierarchy Centralized
- Power Efficient Gathering In Sensor Information System

- Threshold Sensitive Energy Efficient Network Protocol
- Adaptive Periodic Threshold Sensitive Energy Efficient Sensor Network Protocol

2.1.10. Packet Scheduling in WSNs

Prioritizing WSN applications requires carefully planning packet scheduling at the sensor. The scheduling of packets can be based on data delivery deadlines, packet priorities, packet types (real-time or non-real-time), and queue numbers. The technique of giving packets the desired priority order is known as packet priority scheduling. The deadline of a packet, the nature of data, or the quantity of queues can all be used to determine a packet's priority. Prioritizing packets will shorten the delivery time for non-time-critical applications while reducing the delay for time-critical applications (Jain et al., 2014).

According to (Jandaeng et al., 2011), packet scheduling at SNs is essential because it ensures the delivery of different types of data packets based on their priority and fairness with the least amount of delay. Researchers tried the following things to accomplish these goals:

Table 2.1 Packet scheduling schemes comparison in WSNs

Sched. Scheme	Primitive/non-primitive	Strategy	Queuing Strategy
FCFS	non-primitive	Packets coming into queue are schedule in order to arrive	Single level queue
SJF	primitive	Queued packets are scheduled based on completion time	Single level queue
EDF	primitive	Packets with earliest deadline will be scheduled first	Single level queue with high priority with earliest deadline
MP	non-primitive	Incoming packets are queued in to multiple level of queue	Multiple level queue with varying priority
DMP	Primitive when real time packets arrive while non-real time packets are being executed	Dynamic queue allocation method based on their importance	Three level queue with high priority to real time packets

2.2. Medium Access Control Protocols for WSNs

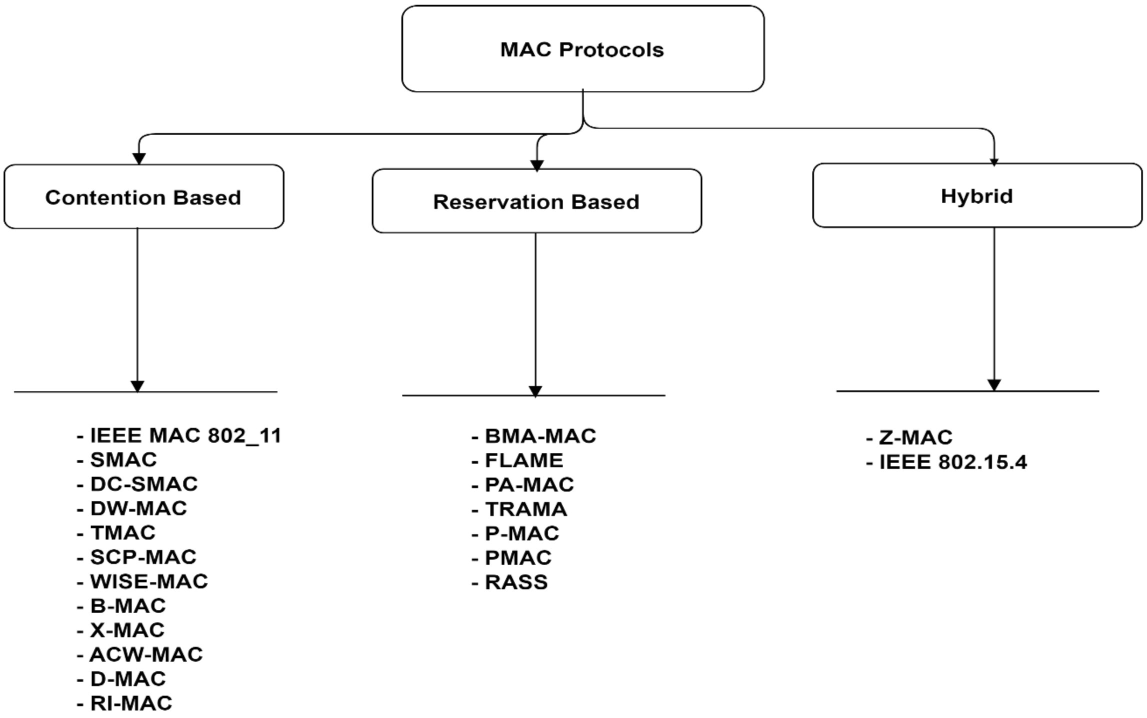


Figure 2.5 Categories of MAC with their examples

MAC protocols in WSN may be divided into two fundamental categories: contention-based and contention-free protocols. But certain protocols, known as hybrid protocols, have characteristics that are both contention-based and contention-free.

2.2.1. Contention Free and Hybrid MAC Protocol

New Sleep/Wake Scheduling Technique for WSNs Based on Data Fusion, an extended resting duration for the cluster head is described in (Song Feiyan, 2016). The strategy makes advantage of a data fusion window and recalculates the best times to sleep. The sum of the cluster's initial child node count, transmission speed, and sensing latency determines the starting value of the data fusion window. Periodically, the head checks the number of children stored there to the number of children in the node tree. If there are fewer children than there were before, the data fusion window will grow narrower. The decrease in size is a result of the cluster's current kid count, transmission speed, and sensing latency. The cluster's sleep time is changed, and the subsequent fusion window is altered, if there is a dead node. As a result, the cluster head sleep time will be modified in accordance with the following fusion. Using an adoptive data fusion window, this plan lowers cluster head energy consumption. Numerous studies have been conducted in this area, and I have attempted to compare some of them using the table below.

Table 2.2 Contention free MAC protocols comparisons

Parameters	Working principle (strategy)	Advantage	Disadvantage
Routing Protocols			
RASS	Cluster head polls cluster members and adjust sleep schedule based on data sending rate nodes.	It is energy efficient since SNs with low sending rate become sleep.	It does not consider priority of received data since node with low sending rate may have more important data.
TRAMA	Use Distributed slot selection algorithm to select one transmitter in two hop neighbor. Bitmap is used to specify intended receiver of the schedule.	Save energy since nodes which are not transmitting and receiving goes to sleep. Throughput is better compared to contention based protocols.	High delay since it increases percentage sleep time. It is suitable for non-time critical application which requires energy efficiency and higher throughput.
P-MAC	Grade division scheduling algorithm is used to divide network around the sink. Pipelining is used to reduce network latency.	Better throughput and average power saving due to implementation of pipeline concept.	Exchanging grade information among nodes creates additional overhead.
ZMAC	Hybrid MAC protocol which alternate between CSMA and TDMA scheme based on network contention. In low contention CSMA is used	Hybrid protocols overcome most of the problem of single scheme based protocol. Perform well both in low and	Switching from CDMA based scheme to TDMA based scheme result in overhead of slot

	while TDMA is used for high network contention.	high traffic condition.	assignment and clock synchronization
PMAC	Pattern generation scheme is used to generate pattern that determine when to sleep and wakeup. Supper time frame has two periods: Pattern generation period and pattern repeat period.	When traffic load is light nodes spent most of their time in sleeping to save energy.	It needs strict synchronization and knowledge of network topology. More collision occur during exchange of pattern

In the hybrid media access control protocol, Z-MAC is made for WSNs (Arshad et al., 2013). A hybrid MAC protocol called Z-MAC alternates between TDMA and CSMA as necessary. Based on network congestion, the decision is made to switch between CSMA and TDMA. The majority of single scheme protocol's drawbacks are solved by hybrid protocols. CSMA is the best option when there is little network conflict, but it might have low throughput when there is a lot of traffic. On the other hand, TDMA can effectively schedule nodes and sustain high channel usage in situations of strong network congestion, but slots are lost in light traffic situations. When there is little channel contention, ZMAC uses CSMA; when there is much channel conflict, ZMAC switches to TDMA.

2.2.2. Contention Based MAC Protocols in WSN

Multiple SNs can share a radio channel in a WSN without previous coordination (or channel reservation) thanks to contention-based communication. There are a number of protocols based on contention, but they all share a channel access policy that is based on competition. The node tries to get access to the channel whenever it wants to send a packet. The main issue with these protocols for delivering QOS is that access to the channel cannot be assured in advance. Contention Based MAC Protocols for WSN include the following:

In (Shaiful & Mohammed, 2010), authors evaluate the efficiency of CSMA and IEEE 802.11. CSMA is the name of one of the commercial wireless local area network technologies. When a packet has to be delivered, each sensor node in CSMA employs the RTS/CTS/ACK method to sense the medium. Each sensor node first plays the material for the DIFS time period. The node begins sensing for a random back-off period after sensing for a DIFS time interval and determining if the medium is idle. After identifying the empty channel, RTS/CTS transmission started. After the back-off timer has decreased to zero, the source node sends an RTS packet. Delivery of CTS packets is postponed by the receiver node for a SIFS (short inter-frame spacing) period of time. After a SIFS period, the receiver answers with a CTS packet. If CTS is not received by source node (the node that sends the RTS packet), waits for the CTS time-out period and retransmits the RTS packet. CSMA uses four-way handshaking techniques with RTS/CTS/ACK and data. The use of four-way handshaking procedures has the benefit of preventing hidden terminal issues and reducing the amount of energy needed to retransmit actual data. Hidden terminal problems are a prevalent issue in CSMA, and the periodic sleep/wake scheduling approach is not used. CSMA requires more energy due to its extended idle listening period and is therefore inappropriate for wireless sensor nodes with limited energy.

The authors of ((Daesuk et al., 2008) suggest a communication protocol based on converge cast. Within sensor networks, convergence-cast is the most often seen communication pattern. The pathways from potential sources to the sink in data collection trees can be thought of as unidirectional. Very low latency and energy efficiency are the two objectives of DMAC. The Slotted Aloha method, which DMAC is an improvement of, assigns slots to groups of nodes depending on the data collection tree. A node's receive period coincides with the transmission and contention periods of all of its child nodes. The nodes that follow one another in the data transmission line are given consecutive slots, which reduces latency. In comparison to other techniques of allocating sleep/listen periods, DMAC obtains the lowest latency by using this methodology. The network's latency becomes important in specific circumstances; in these situations, DMAC can be a good fit. The drawback of DMAC is that it ignores collision avoidance strategies that could be applied when several nodes trying to connect with a single node at the same time have the same schedule. In SNs with event triggers, this scenario could occur. Additionally, it's possible that the data transmission pathways won't be known in advance, which may make it impossible to create a data gathering tree.

It is suggested in (Sun et al., 2008) to use demand wake-up MAC (DW-MAC). Duty cycle is frequently employed in WSNs to decrease energy waste caused by inactive listening, however duty cycle also adds extra delay to packet delivery. A variety of tactics have been used to shorten this wait, however they primarily work when there isn't much traffic. WSN may commonly experience high traffic loads as a result of broadcast or converge-cast traffic. The authors propose DW-MAC, a novel MAC protocol that allows nodes to wake up on demand during an operational cycle's sleep phase while also avoiding data transmissions from interfering with their intended receivers. DW-MAC employs a low-overhead scheduling algorithm. This demand wakeup adaptively enhances effective channel capacity as traffic load increases over the course of an operational cycle, allowing DW-MAC to provide minimum delivery delay under a variety of traffic loads, including both unicast and broadcast traffic.

Asynchronous duty cycles are used by the Berkeley MAC protocol, often known as B-MAC (Alfayez et al., 2015). Each node in the B-MAC has its own scheme for scheduling duty cycles. The preamble that Node provides with each data packet must be longer than the receiver's resting time in order to guarantee that the receiver is in wake-up mode. A node samples the media only when a preamble has been identified while it is in a wake cycle. B-MAC has reduced power use, throughput, and latency; nonetheless, overhearing and the lengthy preamble are significant downsides.

(Henna, 2012) proposes a receiver-initiated media access control protocol. Reduced power consumption, increased throughput, and a greater packet delivery ratio are all achieved by RI-MAC using the receiver initiated method. Similar to B-MAC, each node has its own independent duty cycle scheduling. The primary difference between RI-MAC, B-MAC, and X-MAC is that, with RI-MAC, the sender stays in active mode until the intended recipient is ready and the message delivery process starts. By transmitting a beacon frame, the receiver will let the sender know.

In addition to the studies mentioned above, there are several others being done in the field of competition-based MAC protocols for wireless sensor networks. Wise-MAC, B-MAC, RI-MAC, and the remainder of our theses discuss Contention-based MAC protocols that are made for wireless sensor networks to improve energy efficiency, delay, and throughput.

2.3. Working Mechanism of MAC Protocols

(Wei et al., 2002) proposed the SMAC protocol for WSN. Periodic sleep wake-up, which enables nodes to sleep and send data while they are simultaneously listening to the channel, is the finest feature added by SMAC. Periodic sleep and listening SMAC, especially when network traffic is light, lowers the energy consumed for idle listening, a significant energy consumer in WSN. In order to save energy, Node turns off its radio when it is sleeping. The duty cycle is followed by SMAC to place the nodes in the listen and sleep phases. Through periodic neighbor detection, broadcasting SYNC, and neighbor list upkeep, SMAC specifies a full synchronization system.

The duty cycle is the user-adjustable ratio of the listen period to the frame length for a certain application. All nodes in a network have the same fixed frame length, even though the listening period may change. Every schedule that SMAC nodes adhere to is managed by a timer that has the ability to reschedule when the current one is complete. A timeline for nodes must be at least one page long. Each frame in SMAC has a checking point that serves as its expiry time. SMAC will make a choice for the following period at each checkpoint.

SMAC also has additional characteristics like message forwarding and overhearing prevention. Nodes can overhear packets that are not intended for them, which is known as overhearing. SMAC sends nodes from sender and recipient neighbors to sleep in order to prevent overhearing. A message may be transmitted as a single packet or a fragment. Only one RTS or CTS packet is used to send each piece of a lengthy message when it is transmitted in bursts using SMAC. However, the receiver must send ACK packets each time it gets those fragments. The sender increases the duration of its broadcast if a packet is lost.

S-MAC uses both physical and virtual techniques to detect carriers. If both virtual and physical carrier sensing reveal that the medium is free, it is referred to as being free. The physical layer conducts physical carrier sensing to determine the current radio state each time radio starts or stops receiving or sending, and the physical layer tells the MAC layer. Broadcast packets and unicast packets are the two different types of packets used by S-MAC. Prior to transmission, broadcast packets are sent without the use of RTS or CTS. The RTS/CTS/DATA/ACK sequence is used by unicast packets to deliver data. After carrier sensing, a node must wait for a period of time known as DIFS time before it can begin transmitting data. The minimal idle time required by a medium for contention-based MAC protocols is known as DIFS.

SMAC's sleep/wake scheduling lowers energy usage, however in a multi-hop scenario, it causes considerable delay in the transmission of data packets. The fundamental principle of adaptive listening is to allow each node to continue sending data packets after the current transmission has ended. SMAC is possible to grant each node an extended data period for data packet transmission or reception thanks to this capability.

Low duty cycle decreases SN energy usage but increases delay. When a node receives a data packet from a downstream hop, it will delay delivering that packet until the subsequent hop's nodes' next listening period. When there is a multi-hop communication network, this results in a significant latency. Each node in a network adheres to a defined duty cycle, which is another disadvantage of SMAC. The minimal duty cycle operates better at conserving SN energy when network demand is low, whereas the maximum duty cycle performs better when network load is high. Since traffic load might fluctuate over time, it is impossible to find a fixed traffic type in a wireless sensor network. Researchers tried to find a remedy for this issue by adjusting the duty cycle to account for network traffic. The following characteristics apply to SMAC:

2.3.1. Periodic Sleep Listen

The periodic sleep listening of SMAC is its most well-known feature. It comprises sleep, data, and sync time. A brief window of time known as sync time permits sync between nodes. The real data is delivered during the data time, which is nearly twice as long as the sync time. Nodes slumber at sleep time to reduce energy consumption that may occur during inactive listening. The sleep period lasts a longer time than the data period and sync period. After sending RTS and receiving CTS packets, nodes are then prepared to send and receive data during the data period. By using scheduling and contention techniques, SMAC aims to cut down on energy usage and prevent collisions. Nodes awaken during the listen phase to listen to the channel and connect with other nodes that need to transmit or receive packets in order to cut down on the amount of time spent doing so. However, node will go to sleep by shutting off their radio throughout the sleep time.

2.3.2. Duty Cycle and Synchronization

Duty cycle, also known as frame duration, is the proportion of listen period to entire sleep listen period.

$$duty\ cycle = \frac{Tlisten}{Tframe} \tag{2.1}$$

To regulate the length of the sleep time, the duty cycle may be changed from 10% to 100%. SYNC period and DATA period are the two components of the listen period. The period during which SYNC packets are transmitted is known as the SYNC period. SYNC packets are broadcast packets that can address the issue of SN sync. Data packets, including control packets like RTS and CTS, are exchanged during the DATA period between nodes.

Table 2.3 SMAC frame format

SYNC	DIFS	CW for SYNC		durSYNC	Guard Time		
DATA	DIFS	CW for data	durCtrl	Proc Delay	SIFS	durCtrl	Guard Time

Table 2.4 SMAC SYC and DATA packet frame format with description

Fields	Description
DIFS	In order to employ contention-based MAC protocols, the medium must have a minimum idle time known as DIFS (DCF inter-frame space). In the DCF node that must transmit the data frames, channel activity is monitored until DIFS. The node still waits for a random back-off delay before each transmission after detecting an idle DIFS.
DATA CW	CW for DATA packets, utilized for data packet transmission.
SYNC CW	CW for the SYNC packet, which is used to synchronize different nodes.
SIFS	Short (ms) interface space. Prior to transmitting a CTS or ACK packet, it is utilized. It takes care of each packet's processing delay.
Guard time	ms of guard time at the conclusion of each listen interval.

Packet of SYNC by broadcasting SYNC packets, neighbor nodes in SMAC exchange schedules. Over the course of the SYNC period, SYNC packets are transmitted.

Table 2.5 SYNC frame format

Fields	Description
type	Flag indicating this is a sync packet
Length	Fixed size with 9 bytes
srcAddr	ID of the sender
syncNode	ID of sender‟s synchronization node
sleepTime	Sender‟s next sleep time from now
state	Indicate whether the node change schedule recently
crc	Cyclic redundancy check

To store its own schedules and the schedules of its neighbors, each SMAC node keeps at least one schedule table. A main schedule is one that is created by the node itself, while secondary schedules are those created by other nodes in the table. The node might not have a secondary schedule if every member of its neighbor follows the same schedule.

Table 2.6 Field definition of schedule table

Fields	Description
txSync	Flag showing need to send SYNC
txData	Flag showing need to send data
numPeriods	Counter for sending sync period
numNodes	Number of nodes on this schedule
syncNode	The node which initialize this schedule
chkSched	Flag indicating need to check numNodes

2.3.3. Neighbor List and Carrier Sense

A SMAC-using node keeps track of its neighbors in a neighbor list. Using the user-adjustable SMAC option, the maximum number of neighbors in a neighbor list may be changed. SYNC packet exchange establishes a neighbor list. Unicast data is transmitted once the destination node's inclusion in a neighbor list has been verified.

Table 2.7 Neighbor list field description

Fields	Description
nodeID	ID of this node
schedID	ID of schedule that is followed by this node
active	Flag indicating this node is active recently
state	Flag indicating this node has changed the schedule

SMAC uses both physical and virtual techniques to perform carrier sensing. If both virtual and physical carrier sensing reveal that the medium is free, it is referred to as being free. The physical layer conducts physical carrier sensing to determine the current radio state each time radio starts or stops receiving or sending, and the physical layer tells the MAC layer.

2.3.4. Collision and Overhearing Avoidance

The RTS/CTS technique used by SMAC, which is described in DCF, can shorten collision times and address the so-called "hidden terminal problem." The sender and receiver should trade RTS and CTS packets before the real DATA packet. Broadcast packets and unicast packets are the two different types of packets used in SMAC. Prior to transmission, broadcast packets are sent without the use of RTS or CTS. The RTS/CTS/DATA/ACK sequence is used by unicast packets to deliver data. A node cannot begin transmitting data right away after carrier sensing because it must wait for a period of time known as the DIFS time. The minimal idle time required by a medium for contention-based MAC protocols is known as DIFS. According to (Zou et al., 2012), a node computes a random back-off time with eq. 2.2 to prevent collision.

$$\text{Back-off} = \text{slotTime} + \text{Random}\ (0,CW) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\ 2.2$$

During the Data time, transmission and receiving of data are carried out. Radio bandwidth and CW size are two physical and MAC layer characteristics that determine the length of the data period. RTS/CTS/DATA/ACK packet transmission and collision are both possible. For instance, if a collision occurs while two neighbor nodes are transmitting RTS packets simultaneously, SMAC will retransmit the RTS packet after a random back-off period. After waiting for the CTS expiration time, during which the sender should receive CTS following the delivery of RTS packets, those nearby nodes become aware of the collision.

In wireless sensor MAC protocols that rely on contention, overhearing is a significant energy drain. When a node receives packets that are meant for another node, it happens. Allowing each node to listen in on all of its neighbors' communications is the most effective technique to accomplish overhearing. When there is little or no network traffic, this functionality is inefficient in terms of energy use. Every time SMAC nodes receive CTS or RTS packets that are meant for the other nodes, they fall to sleep to prevent overhearing. In SMAC, the sender's and receiver's neighbors essentially fall asleep.

2.3.5. Message Passing and Adaptive Listening

The message passing mechanism, which involves breaking up lengthy messages into smaller units (called fragments) and sending them to the recipient, is another aspect of SMAC. SMAC only transmits one RTS packet and one CTS packet for each fragment, although the sender anticipates receiving an ACK packet for each fragment.

To decrease the delay of scheduling sleep and waking in SMAC, adaptive listening is suggested. SMAC's sleep/wake scheduling lowers energy usage, however in a multi-hop scenario, it causes considerable delay in the transmission of data packets. The fundamental principle of adaptive listening is to allow each node to continue sending data packets after the current transmission has ended. SMAC is possible to grant each node an extended data period for data packet transmission or reception thanks to this capability.

Authors (Wang et al., 2013) suggest an adaptive duty-cycle method by substituting a priority discriminant function for the fixed duty-cycle of SMAC. Due to this, nodes with more packets have priority over nodes with fewer packets when it comes to channel access. The fundamental concept underlying dynamic duty-cycle is that it doubles when network traffic is relatively high, allowing nodes to listen for longer and minimizing packet loss, while it is cut in half when network traffic is relatively low. Through the use of a mathematical model, the authors demonstrate the relationship between duty-cycle and frame length. The total of the listen period and the sleep period is the frame length. The relationship between duty-cycle and T frame is inverse; reducing the duty-cycle results in an increase in the T frame value, which results in more delay. In order to dynamically change network duty-cycle, DC-SMAC can forecast network traffic.

Under varying traffic loads, T-MAC is proposed to address S-MAC's drawback (Dam & Koen, 2003). When no activation event has place between nearby nodes within the time threshold TA, a node following T-MAC terminates its listen period. As a result, TMAC takes the network's varying load into account. The periodic frame timer, any data being received over the radio, the end of its own data packet's transmission or an acknowledgement, and the overhearing of CTS or RTS are the activation events. A minimum quantity of passive listening each frame is described by TA. While its neighbors are conversing, the T-MAC node won't go to sleep since it could end up being the recipient of following messages. Each node sends out the messages in its queue in bursts; if the transmitting node doesn't get a response within the time interval TA, it will fall to sleep. Since the first frame has already been transferred, this circumstance may occur, decreasing throughput. The node should resend the RTS if it does not receive a response. If it does not, it will wait for two more trails before going to sleep. The total of the contention interval, the size of the RTS packet, and the brief gap between the conclusion of the RTS packet and the start of the CTS packet is more than the value of TA.

The fundamental goal of TMAC design is to shorten the amount of time that nodes may listen to the channel when no packets are being broadcast. In contrast to the other causes mentioned above, idle listening represents the main source of energy loss. By incorporating the activation component TA, TMAC lowers the amount of energy needed for passive listening. TMAC's design has taken into account a number of additional elements in addition to idle listening. These elements consist of:

Collision: When packets are being sent from communication nodes, they may collide with one another at the same moment, wasting the energy needed to send them.

Protocol overhead: T-MAC views the energy needed to send and receive control messages, which are exchanged in the majority of protocols but do not often carry any application data, as overhead.

Overhearing: Airborne communication occurs in a shared medium. Therefore, it may receive packets that aren't intended for it through this communication node. The energy used to receive packets that are not intended for the present node is reduced by avoiding overhearing.

In contrast to the other causes mentioned above, idle listening represents the main source of energy loss. By incorporating the activation component TA, TMAC lowers the amount of energy needed for passive listening. TMAC's drawback is that it has issues with sleep schedules. T-MAC is more prone to poorer throughput than the older SMAC protocol because of its early napping issue.

An adaptive media access control mechanism for contention windows is suggested by the authors in (Weixia et al., 2012). The SMAC protocol can't adjust to changing network traffic, which might cause significant delays and possibly excessive power use. Due to the shortcomings of the SMAC back-off mechanism, ACW-MAC alters both the contention window and the RTS frame. The back-off method suggested by ACW-MAC employs a stable contention window as opposed to a continual contention window.

Determine a random back-off time before transmission in the SMAC node and set the back-off timer. When the timer is decremented to 0, the node begins to broadcast. For collision avoidance, random backtracking is crucial. CWinit and CWbasic are two new parameters introduced by ACW-MAC. The initial contention window, abbreviated CWinit, is created by dividing by two the total of the minimum and maximum contention windows. Then CW will be given the value of the first contention window.

$$\text{CWinit} = \frac{cwmin+cwmax}{2} \quad \text{……………………………………………………………………………2.3}$$

Another option introduced by ACW-MAC is called ACCESS, which will be set to 1 if the node successfully accessed the channel and transmitted a packet the previous time. ACCESS will be set to 0 if the node failed to successfully send a packet the previous time. By contrasting the present contention window with the fundamental contention window, the amount of network traffic at any given time is calculated. Then, a recently introduced RTS packet header called CONTENT is used to inform nearby nodes about the present contention condition. The authors use a five-node star topology to assess the performance of SMAC and ACW-MAC. The performance of AODV is assessed as a routing protocol. According to the results of the performance evaluation, changing the CBR interval makes ACW-MAC operate better in terms of average end-to-end latency, throughput, and average energy efficiency.

2.4. Related Works

This section addressed and evaluated the study topics linked to the MAC protocol of WSN's energy efficiency techniques. Based on this thesis study, the pertinent gaps and improvements were also identified. Presumably, a lot of studies that looked at the effects of energy depletion in the WSN protocol, notably in the MAC, largely focused on looking into node listening and sleeping times, which causes nodes to wait longer while they are not doing anything. The majority of studies lacked a method for forcing nodes to sleep while they were still in the middle of their duty cycle.

(Ye et al., 2002) is MAC Protocol for Multi-Hop WSNs. SMAC aims to address the unique challenges of energy efficiency and scalability in WSNs by employing a contention-based approach with duty cycling. SMAC utilizes duty cycling to conserve energy in WSNs. SNs alternate between active and sleep states to minimize idle listening and idle transmission. SMAC incorporates synchronization mechanisms to ensure that SNs operate on the same schedule. This synchronization is achieved through periodic wake-up beacons sent by a designated coordinator node, allowing neighboring nodes to align their duty cycles. SMAC employs adaptive listening to further enhance energy efficiency. Nodes dynamically adjust their listening schedules based on the arrival of wake-up beacons and incoming data packets.

The sensor medium access control protocol for WSNs is being improved in (Teshome, 2018). The MAC protocol's design is expected to incorporate crucial elements like energy economy, reduced latency, greater throughput, and topology change adaptability. With regard to static duty cycles, they suggest enhancing the SMAC protocol. Because of its inflexible nature and inability to respond to network changes, SMAC might have increased end-to-end delays and energy usage as a result of having to retransmit missed packets.

Even though SMAC is a synchronization-based MAC technology, full-time synchronization between WSNs is essentially unattainable. They have created an algorithm that uses the network size and remaining energy of a node to change the CW size and duty cycle of the SMAC protocol in order to solve the issue with the fixed CW size and fixed duty cycle of SMAC. A smaller value of CW for a large number of nodes results in intense computation among SNs, which causes packet loss. This is because CW contains a time slot that each node is required to wait before transmitting or receiving sync packets, data packets, including control packets, and control packets.

Due to its restrictions in terms of processing, storage, and computational capacity, the MAC protocol created for WSNs is entirely different from the MAC protocol used in conventional wireless networks. Enhancing energy efficiency while preserving collision avoidance and scalability is one of the key features. In (Razaque et al., 2018), EE-MAC protocol reduces idle listening, overhearing, and preamble length in an effort to increase energy efficiency and throughput. The EE-MAC combines TDMA and CSMA functionalities. The EE-MAC gets TDMA characteristics to arrange the time slot as a baseline, and CSMA aids in channel congestion if necessary so that it may be used for an unlimited amount of time. EE-MAC seeks to balance all functions while improving energy and throughput efficiency.

The deficiencies in conventional MAC protocols are filled by the Energy Efficient Adaptive MAC Protocol for Mission Critical Applications in WSN protocol (Sakya & Sharma, 2019). This protocol has two suggested algorithms. The first approach, which is priority-based, chooses the node with the highest number of packets in the queue and the most energy out of all the nearby nodes constituting the virtual cluster as the cluster leader. Based on traffic circumstances and the node's remaining energy, the second algorithm use regression approach to determine the duty cycle of the chosen node. Only chosen nodes participate in packet transmission. As a result, there is less packet overload at intermediate nodes.

The authors (Dattatraya & Rao, 2022) described a node cooperation approach in which a sender can help one or more nodes that have a larger medium gain and sufficient residual energy to transmit their data packets to its recipient. They recommended a transmission energy optimization technique to increase the least amount of energy left over after data packet transfers from the sender and its cooperating nodes, hence extending the lifespan of

WSNs. Based on this concept, they proposed a corresponding power-optimized cooperative MAC protocol.

Bitmap Assisted Efficient and Scalable TDMA based is also known as BEST-MAC. This protocol was created using MAC with a schedule. The number of tiny time slots used by the MAC (Kumar & Gangwar, 2023) should not be identical to the number of member nodes in a cluster. It was proposed to employ short addresses to decrease control overheads, and the Knapsack algorithm was used to improve connection usage and shorten node task length.

The summary of some papers is summarized as follows:

Table 2.8 Summary of related works

MAC Protocols	Authors	Working principle	Problem Addressed	Gap(s)
SMAC	Wei, John, and Deborah, 2002	Adaptive listening, Static sleep schedule	Reduce collision and minimize energy consumption	RTS-CTS increase energy, Fixed CW and Fixed duty cycle
EE-MAC	Razaque et al., 2018	Merges of CSMA and TDMA.	Improve energy efficiency and throughput, minimizing idle listening, overhearing.	Switching from CSMA to TDMA result in overhead of slot assignment and clock synchronization.
ADMC-MAC	Sakya & Sharma, 2019	Priority based CH and maximum energy based virtual cluster forming, deciding the duty cycle based on traffic conditions and residual energy.	Improve energy efficiency, while maintaining a collision avoidance and scalability.	As residual energy decreases the chance of node in cluster is decreased which result delay of information.

ESMAC	Teshome, 2018	Adjust c.w based on network size, and duty-cycle based on remaining energy.	Consume less average energy than SMAC, and increase throughput.	High listening time during high remaining energy which increase energy consumption.
HB-CHS for Maximizing NLT and Energy Efficiency in WSN	Dattatraya & Rao, 2022	Proposes a new Fitness based GS with FA, which is the hybridization of GSO and FO algorithm to choose the best CH in WSN.	NLT maximizing by minimizing energy consumption.	Algorithms are not much capable to address several problems such as overhead, and communication delay.
Improved BEST-MAC protocol for WSN using optimal CH selection	Kumar & Gangwar, 2023	Optimized CH selection, back propagation algorithm for data transmission.	Maximize life time and throughput of a WSN.	Backpropagation need additional appropriate initialization, and computational intensity

2.5. Summary

Various research on how to conserve energy and prolong the life of nodes have been examined in a number of articles. Both the MAC and routing layers have been worked on utilizing various strategies to reduce energy consumption and improve network performance in WSNs. By carefully examining their work, notably the portions on the MAC protocol, we were able to identify holes. The focus of this thesis work is on MAC protocols for WSNs, and we made an effort to make the ESMAC protocol more energy-efficient. I have concentrated on resolving issues with energy usage by using the ESMAC protocol's basic idea and incorporating various approaches from other papers. SLT, node distance-based clusterization, and remaining energy-based cluster head selection were created using the remaining energy-based adjustment duty cycle. The suggested EES-MAC will be discussed in the next chapter.

CHAPTER THREE

PROPOSED SOLUTION

One of the most well-known ways to conserve energy in WSN is by designing the MAC protocol effectively. Periodic sleep wake-up, which SMAC adds, has one very obvious disadvantage in relation to the static duty cycle. Owing to its fixed nature and inability to respond to network changes, SMAC has the potential to have increased end-to-end delays and energy usage owing to the need to retransmit lost packets. The ESMAC method used the network size and remaining energy of a node to change the contention window size and duty cycle of the SMAC protocol in order to solve the issue with the fixed contention window size and fixed duty cycle of SMAC. In this paper, the ESMAC protocol for WSNs is enhanced. The MAC protocol's design is expected to incorporate key elements, including ensuring energy efficiency, minimizing latency, increasing throughput, and being adaptable to changes in topology. Since the nodes wait for a time to finish their duty cycle for different reasons while they don’t perform any activity but consume energy, the ESMAC protocol suffers from the same flaw in terms of energy conservation. This feature of ESMAC makes us study how to save energy while improving throughput and minimizing latency. Even though ESMAC is a synchronization-based MAC system, full-time synchronization between wireless sensor nodes is essentially unattainable.

To fix the problem related to energy consumed by idle listening because nodes wait time to finish their duty cycle without doing any activity in the pre-designed duty cycle of MAC protocols, we developed an algorithm that sets more sleep time for nodes than listen time and utilizes distance between nodes and the remaining energy of a node to adjust clusterization. Since large amounts of sleep time are set for nodes, our algorithm creates a high amount of delay. To solve the delay problem, we had to write another algorithm that creates clusters based on inter-cluster distance and selects cluster heads for each group based on the energy they have. We run simulations with various node counts and duty cycles to validate our findings. To choose the cluster head for each group, we calculate the inter-cluster distance using the SNs remaining energy. The chosen node then engages in its group member activities while in sleep mode. The upgraded sensor medium access control protocol now performs better in terms of throughput and latency thanks to the combination of these two elements.

3.1. Network Model

SNs are distributed in an 801x500 network field, with changes in number of nodes starting from 6, 10, 30, 40, 50, 60, 70, 80, 90, and 100, and by changing duty cycle from 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. According to residual energy and inter- group distance, the suggested work splits the network's SNs into a number of groups and allocates a single efficient CH to each group. The CH is a node that gathers, aggregates, transmits, and executes operations for nodes in his cluster that are sleeping, and then delivers it to the sink.

3.2. High-Level Design for the Proposed Work

The major goal of the proposed study is to create MAC protocols based on the ESMAC protocol that increase throughput while being energy-efficient. The high-level architecture for the proposed work is based on ESMAC protocol characteristics as well as new features that we suggest as enhancements to the ESMAC protocol for WSNs.

There are the following three phases in EES-MAC:

i. *Setup phase*

First of all, the necessary requirements are met for the work to be done. These include determining the residual energy and contention window, adjusting remaining based on duty cycle, and selecting CH based on the inter-cluster distance and remaining energy.

ii. *Decision phase*

The nodes identify their current status during this phase and decide whether to proceed to the following step or sleep. Are there any of the following: a sync period, a receive sync, a data period, and a receive CTS? The alleged are finished.

iii. *Final phase*

Depending on the above phase, the nodes perform the next task or go to sleep. Broadcasting sync packets, inserting received schedules as secondary schedules, broadcasting RTS packets, and sending data are the activities in this phase.

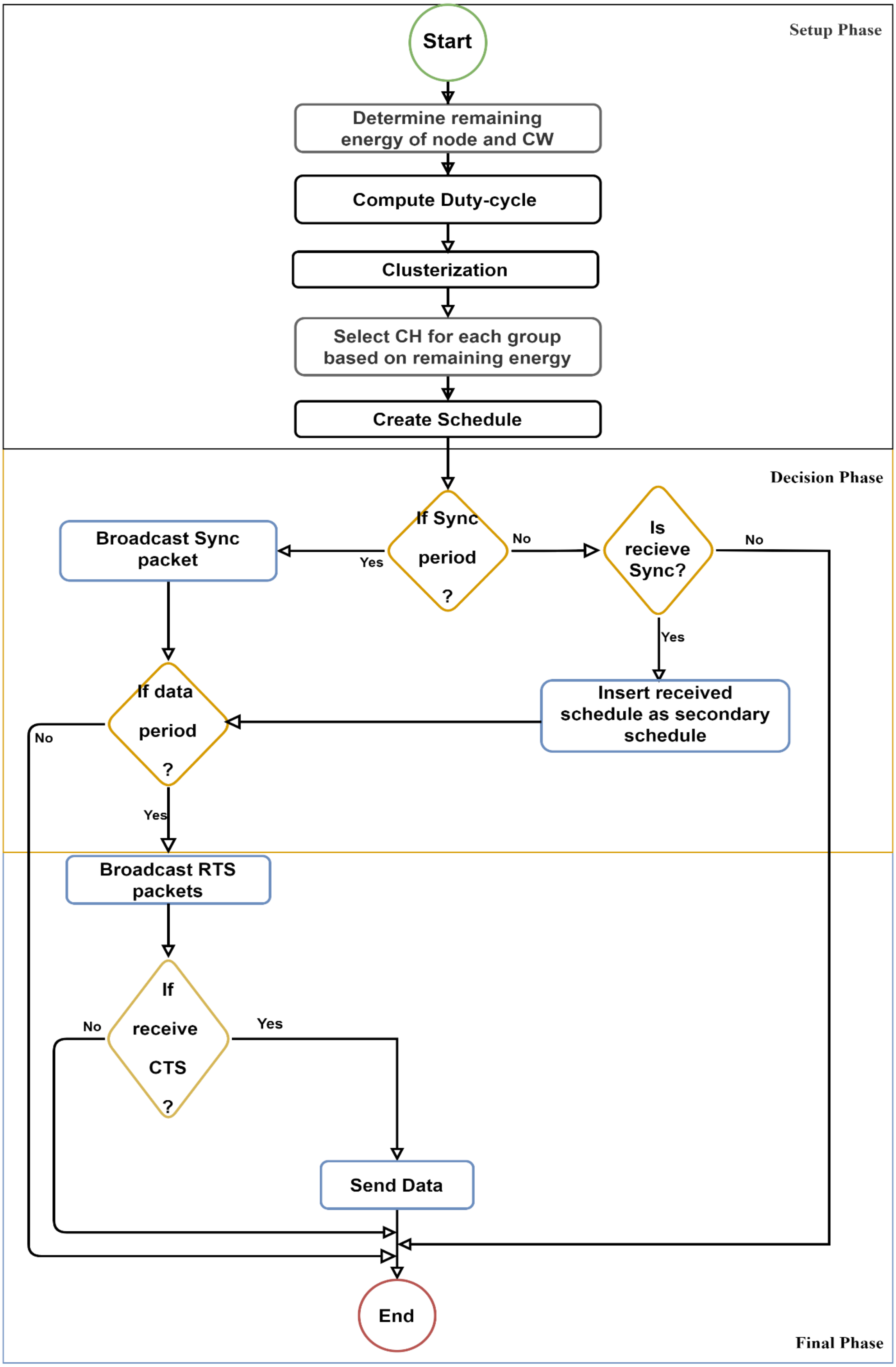


Figure 3.1 Flow chart of EES-MAC

3.2.1. Description of the Flow Chart

The ESMAC capabilities are maintained in EES-MAC, but some new features are added by employing customizable SLT and clusterization based on the amount of energy left in each node in the network to change the SNs' sleeping times and choose CHs for each group.

The stages of EES-MAC are:

a. Determining remaining energy of nodes and compute duty cycle

This feature, which was also copied from (Teshome, 2018), is used in a new method in this work to compute the cluster head for each group and to change the nodes' duty cycle. The nodes of a network assess their remaining energy to predefined standards. When the SNs' energy drops below the specified threshold, both the cluster heads of each group and the SNs' duty cycles are modified. A sensor node becomes less active and receives a new cluster head when its energy drops below a particular threshold. SNs may sleep for longer stretches of time and use less energy by reducing their duty cycle. Energy conservation is predicated on the length of the sleep period in contention-based MAC protocols that use duty cycles:

$$ES = \frac{Tsleep}{Tframe} \quad \text{3.1}$$

Length of Frame (Tframe) is the complete listen sleeping cycle

$$\text{Tframe} = \text{Tsleep} + \text{Tlisten} \quad \text{3.2}$$

From equation 3.2, one can calculate sleep time (Tsleep) as follows:

$$\text{Tsleep} = \text{Tframe} - \text{Tlisten} \quad \text{3.3}$$

Equation 3.3 is substituted in Equation 3.1 as follows:

$$ES = \frac{Tframe - Tlisten}{Tframe}$$

$$ES = \frac{Tframe}{Tframe} - \frac{Tlisten}{Tframe} = 1 - \frac{Tlisten}{Tframe} \quad \text{3.4}$$

Duty-cycle is computed as a follows:

$$duty\ cycle = \frac{Tlisten}{Tframe} \quad \text{3.5}$$

When eq. 3.5 substituted in eq. 3.4, the energy savings of a node are:

$$ES = 1 - duty\ cycle \quad \text{3.6}$$

We may infer from eq. 3.6 that the energy savings of SNs rise when a node's duty cycle is decreased.

Due to SNs' longer sleep duration, a reduction in duty cycle results in greater energy savings. We suggest the SLT, cluster chiefs that are elected for groups of nodes based on eq. 3.6, and an adaptive duty cycle that takes into account SNs' remaining energy. Following deployment, SNs experience energy loss as a result of a variety of phenomena, including idle listening, overhearing, collisions, data routing, etc. These elements lead to uneven energy consumption among sensors, with both less and more energy left over. The suggested EES-MAC protocol modifies the duty cycle and SLT using algorithm 3 to lessen the likelihood of this circumstance occurring. According to specific physical layer characteristics in SMAC, frame length is fixed. Thus, the listening time for a certain duty cycle may be calculated as follows:

$\mathrm{Tlisten} = dutycycle * Tframe$

The following equation will be used to calculate sleep duration:

$\mathrm{Tsleep} = \mathrm{Tframe} - \mathrm{Tlisten}$

Think about the scenario when the duty cycle is 20%. Let's assume that each frame lasts 70 seconds. You may calculate the listening duration as follows:

$Listen_{time}$ = 20/100*70 = 14 seconds

$Sleep_{time}$ = 70 – 14 = 56 seconds

In our system, a sensor node's duty cycle increases to 0.80 times its initial duty cycle if its residual energy falls below 0.80 of its initial energy. Duty cycle in the above illustration becomes: 0.80 * 20 = 16 %.

$\mathrm{Listen}_{\mathrm{time}}$ = 16/100 * 70 = 1142/100 = 11.42 seconds

$\mathrm{Sleep}_{\mathrm{time}}$ = 70.00 – 11.42 = 58.58 seconds

According to the aforementioned example, the suggested approach will allow for greater sleep time than the ESMAC protocol. EESMAC is able to conserve energy by sleeping nodes longer when their energy falls below certain thresholds (i.e., 0.80, 0.60, or 0.30) of their starting energy.

b. Determining CW

Similar to ESMAC, which increases the throughput of WSNs that use the sensor MAC protocol, each node in the network decides its contention window during the first phase based on the number of nodes in the network. When the CW matches the number of network nodes, SMAC performs better in terms of throughput and delay.

c. Creating cluster and selecting cluster head

The net is divided into rounds based on distance they have with together i.e. clustering into different group based on distance they have, each of which has a predetermined length. At the beginning of each round, the nodes in the network compete to become cluster heads. During the first part of the round, each node calculates its election probability based on its remaining energy, as well as a constant known as the "competitiveness factor". The competitiveness factor is used to control the number of cluster heads in the network and is typically set to a value between 0 and 1. The election probability for each node is calculated as follows:

$$\mathbf{E\ (i)} = \frac{\mathbf{R\ (i)}}{\mathbf{T\ (i)}} \qquad 3.7$$

Where:

- ✓ E (i) is the election probability of node i
- ✓ R (i) is the residual energy of node i and
- ✓ T (i) is the total remaining energy of all nodes within communication range of node i.

Once all nodes have calculated their election probabilities, they broadcast their probability values to the network. Each node sets a probability threshold based on the highest probability value it receives. Nodes with probabilities higher than the threshold become CHs, while the remaining nodes become regular nodes and join the cluster headed by the nearest CH. It also has a rotating cluster head mechanism, which alternates which node serves as the group's CH to further manage energy usage. This helps prevent the energy of a single node from running out too soon and equally divide the energy usage throughout the nodes.

The CHs are accountable for aggregating the data received from their member nodes and sending it to the base station or sink node as well as facilitating every activity during the sleeping time of other group members. For example it takes ACK from receiver node at sleeping time of sender and send back it to sender at its wakeup time to conserve energy.

d. Creating schedule based on CW and remaining energy of nodes

Each node chooses its own sleep and listening schedule based on CW and available energy. A SYNC packet will be used to broadcast the prepared schedule to the nearby nodes. To synchronize the sleep and awakening schedules of nearby nodes, a sync packet is employed.

When receiving a schedule, neighbor nodes will attempt to compare it to their own timetable. If the schedules line up, it will acknowledge the sender; if not, it will add the received schedule to its schedule database as a secondary schedule. A SN requires a sense carrier throughout the listen time before sending or receiving packets. Data packets and sync packets are also used during this time. In ours, the number of nodes in a network decides the CW, which will be less for a small number of nodes but greater for a high number of nodes.

e. Broadcasting RTS/CTS packets

RTS and CTS packet broadcasting serve the same purpose in the MAC protocol. The MAC protocol transmits RTS in order to transmit data from the source to the destination. Whenever it receives RTS, the node will send CTS broadcast packets to the sender node. The node will search for any nearby nodes that are transmitting or receiving if none of its neighbors are currently doing so. To avoid collisions during data transmission, all adjacent nodes that receive CTS broadcast packets will fall to sleep. However, the RTS broadcast packet transmitter won't fall asleep. The sender waits for a certain back-off period after receiving the CTS packet before sending the real packet.

f. Sending Actual Packet

Actual data will be transmitted between neighboring nodes according to the ESMAC protocol at this phase. The ESMAC protocol uses message passing to convey real data across nodes during this phase, which functions as the data sending phase of the protocol. The real data is broken up into little pieces and transmitted in bursts when using ESMAC's message passing mechanism. Each fragment of the same data packet must be acknowledged separately by the recipient since ESMAC only transmits one RTS and only receives one CTS for the whole data packet.

3.3.The Algorithm of Proposed EES-MAC Protocol

Algorithm *1: Algorithm of proposed system*

```
Begin
Determine the node's remaining energy
Create a group and choose the cluster head using Algorithm 2.
Use Algorithm 3 to calculate duty cycle.
Determine CW that fits the size of the present network.
Generate the main schedule
Create the SYNC packet
If the SYNC period, then
        Broadcast SYNC packet
    If a SYNC packet is received, then
        Check the schedule against the main schedule
    If schedule match then
        Adhere to the present schedule
    else
        Insert current schedule as secondary schedule
Else If Data period then
        Send RTS
    If receive CTS then
        Send data
    else
        Go to sleep
Else
    Go to sleep
End
```

Synchronization=SYNC, Request To Send=RTS, Clear To Send=CTS, Contention Window=CW

To effectively manage energy resources, the algorithm starts by figuring out how much energy each WSN node has left. The nodes create groups and choose cluster heads using a predetermined procedure once the energy levels are known. The program then calculates the duty cycle for each node, figuring out how much time it should spend awake and asleep. The method tries to save energy and increase the network's overall lifespan by optimizing the duty cycle. To ensure efficient data transmission, the algorithm adjusts the CW size to match the current network size. The CW affects the back-off mechanism and collision avoidance, improving the network's overall performance.

A primary schedule is then generated, which specifies the time slots for data transmission and reception within the network. This schedule helps in coordinating communication activities and ensures that nodes are synchronized in their operations. During the SYNC period, a SYNC packet is broadcast to all nodes, enabling them to align their activities with the network's synchronized timing. This synchronization process helps in avoiding conflicts and maintaining efficient communication.

If a SYNC packet is received, the algorithm verifies the received schedule against the primary schedule to ensure synchronization. If the schedules match, the algorithm proceeds with the current schedule for data transmission and reception. If there is a mismatch, the algorithm incorporates the current schedule as a secondary schedule, indicating a need for further synchronization adjustments. During the data period, the algorithm initiates data transmission by sending an RTS signal. Upon receiving a CTS signal in response, the algorithm proceeds with sending the actual data. A collision or interference is indicated by the absence of a CTS signal, unless the algorithm instructs the node to enter sleep mode.

The program then instructs the node to enter a sleep state, conserving energy until the next planned activity, if it is not the SYNC or data period. This sleep setting cuts down on wasteful power use while you're not doing anything.

Algorithm 2: Algorithm for cluster formation and select cluster head

```
Begin
Initialize the network and set the rlen.
Divide the network into clusters based on the distance between nodes.
    For each node i:
        Calculate the distance to all other nodes in the network.
        Group nodes together into clusters based on a distance threshold.
Repeat for each round:
    All nodes calculate their E (i):
    For each node i:
        Calculate the T(i).
            $T(i) = SUM[R(j)]$
        Calculate the E(i):
            $E(i) = R(i) / T(i)$.
        All nodes broadcast their E(i) to the network.
        Each node sets a probability threshold based on the highest probability
Set the threshold to the maximum E(i)received.
    Nodes decide whether to become CH or regular nodes:
For each node i:
    If E (i) is greater than the threshold,
        Node i becomes a CH.
    Else,
        Node i becomes a regular node and joins the cluster headed by the nearest CH.
For each CH node:
    Aggregate received data from their member nodes.
    Forward the aggregated data to the SN.
    Facilitate activities of group members during their sleeping time.
For each cluster:
    Determine the node with the lsen among the cluster heads.
    Transfer the CH role to the node with the lsen.
Continue with the next round.
End
```

*Round Length=rlen, Election Probabilities = E(i), Total Remaining Energy=T(i), Cluster Head=CH, Sink Node= SN, Lowest Energy=lsen, R*esidual energy of node i=R(i)

The method begins by establishing the round length and initializing the network. In accordance with the separation between nodes, it is separated into clusters. Each cluster is made up of nodes that are relatively close to one another. The method determines the distance between each node in the network and clusters them according to the distance threshold. The algorithm performs the following steps for each round:

a. *Election Probability Calculation:* For each node in the network, the algorithm calculates the election probability. Based on the node's remaining energy as well as the combined residual energy of all nodes within its communication range, the election probability is calculated.
b. *Probability Broadcast:* Each node broadcasts its election probability to the network, allowing other nodes to receive and process it.
c. *Threshold Setting:* Each node sets a threshold based on the highest election probability it receives from the broadcast. The threshold determines the minimum election probability for a node to become a CH.
d. *CH Selection:* Every node evaluates the threshold in relation to its election probability. It becomes a cluster head if its election probability is higher than the threshold. If not, it transforms into a standard node and affixes itself to the cluster led by the closest CH.
e. *CH Responsibilities:* Data from member nodes is gathered by CHs, who then send it to the base station or sink node. They also facilitate activity when other group members are asleep, such as forwarding acknowledgements from recipient nodes to the sender.
f. *CH Rotation:* The program has a system where the cluster head role is rotated across cluster nodes to balance energy usage. The new CH is chosen as the node among the cluster heads with the lowest energy.

The algorithm continues with the next round, repeating the steps described above.

Algorithm 3: Algorithm of duty cycle adjustment

```
Begin
Get dc's primary value
Get sn's remaining energy
If rem > 0.80 * primary energy then
        DC = dc
else if ((rem <= 0.80 * primary energy) && (rem rem > 0.6 * primary energy)) then
        DC  = dc * 0.80
else if ((rem <= 0.6 * primary energy) &&( rem > 0.30 * primary energy)) then
        DC = dc * 0.5
else
        DC = 0.30 * dc
End
```

New Duty-cycle =DC, duty-cycle=dc, sensor node=sn, remaining energy=rem,

Based on the sensor node's remaining energy and beginning energy level, this method is intended to estimate the ideal DC for it. It begins by getting the duty cycle's starting value. It then gets the sensor node's remaining energy. For the algorithm to choose the right DC value, certain requirements must be met.

The DC stays constant and is reset to the beginning value if the remaining energy is greater than 80% of the starting energy. The DC is changed to 80% of the initial value if the residual energy is between 80% and 60% of the initial energy. The DC is set to 50% of the initial value when the residual energy is between 60% and 30% of the initial energy. Finally, the DC is changed to 30% of the initial value if the residual energy is less than or equal to 30% of the starting energy.

This technique tries to optimize energy usage at the sensor node by dynamically modifying the duty cycle based on the remaining energy. In the wireless sensor network, this can assist increase overall energy efficiency and extend the node's operating lifetime.

The network's node count affects the CW in both data and synchronization. Each node in the network competes for a SYNC_CW time slot before sending a SYNC packet to an adjacent node. As soon as they have won, the channel access nodes begin to deliver their SYNC packets. Nodes must perceive the channel within the time frame specified by DATA_CW before sending DATA packets. If the node is successful in winning the channel after carrying out carrier sensing for DATA_CW time slots, it begins transmitting DATA packets. The minimal CW for syn and data packet delivery results in severe competition between SNs, which increases packet loss. Waiting time increases because a node must feel the channel for the entirety of DATA_CW or SYNC_CW in order to deliver a packet. SMAC employs predetermined contention window widths for both SYNC and DATA.

3.4. Hybrid Scheduling Algorithm of EES-MAC

In WSNs, hybrid scheduling is the employment of a variety of packet scheduling approaches or algorithms to strike a compromise between a numbers of goals. It strives to make use of the advantages of various scheduling strategies to enhance network performance and satisfy various needs. We employ the hybrid scheduling technique in EES-MAC for the following reasons, in more detail:

Hybrid scheduling incorporates two or more scheduling methods into a single framework, including round-robin, priority-based, proportional fair, and opportunistic scheduling. The system is able to dynamically adjust to various traffic patterns, priorities, or network circumstances since each scheduling approach is only used when it meets certain criteria or conditions.

Various goals that need to be balanced are taken into account by hybrid scheduling, including fairness, throughput, latency, energy efficiency, and application-specific needs. Based on the unique objectives and restrictions of the WSN, the scheduling algorithm intelligently chooses the best scheduling strategy or combination of strategies. The algorithms are made to change in response to shifting network circumstances and demands. Based on the observed network circumstances, the algorithm may change between alternative scheduling algorithms or modify their settings. Since the method constantly modifies the scheduling strategy depending on observed network circumstances, we have employed adaptive hybrid scheduling in our research. It regularly checks indicators like energy levels, traffic, or channel quality and adjusts scheduling strategies as necessary.

CHAPTER FOUR

SIMULATION AND PERFORMANCE EVALUATION

In Chapter 3, we designed the architecture and algorithm for an energy-efficient version of the ESMAC protocol.

4.1. Simulation Tools for WSN

Applications including temperature, gas, ambient conditions, humidity, etc. are catered for by WSN. Although wireless sensor nodes have a wide range of applications, their resource-restricted nature creates extra demands for maximizing the use of the finite resources of WSNs networks. Researchers from all around the world continue to study the topic of WSNs in an effort to create new algorithms, protocols, and procedures that will increase the effectiveness of WSNs. Since it is not possible to build each algorithm on an SN, utilizing simulation tools to develop and assess algorithms and strategies becomes the best option. The event, medium, environment, node, transmitter, physical, MAC, routing, and application layers make up the WSNs simulator. Wireless sensor networks employ a variety of simulation techniques.

4.1.1. Network Simulator Two (NS-2)

According to the authors (Bakare & Enoch, 2019), NS-2 is an object-oriented simulator consisting of C++ and an object-oriented command language (OTCL). C++ is used to implement varying protocols and extend simulators, while OTCL is used for creating network topologies, configuring simulators, and setting network topologies. It can run on a variety of OSs, such as Linux, Mac OS, Solaris, and Windows. It supports dual output, test-based and graphical-based. Graphical simulation is performed using NAM. The results of the simulation were drawn from a trace file using XGRAPH. NS-2 provides support for different protocols, including MAC-layer protocols such as 802.11, 802.15.4, 802.16, SMAC, etc. We have used the NS2 simulator tools for the following reasons:

NS2 has a larger library of pre-existing protocol models compared to NS3. It has been actively developed for a longer period and, as a result, offers a more extensive collection of protocol implementations. It is a well-documented set of protocol models and examples, making it easier for researchers and developers to understand and use the existing models as a starting point for their simulations. NS2 has a larger and more established user community compared to NS3. This larger community translates into more available resources, including online forums, mailing lists, and tutorials. The well-established community around NS2 provides a helpful environment for knowledge sharing, troubleshooting, and collaboration.

Due to its longer history and widespread adoption, NS2 has a larger body of research literature based on its simulations and evaluations. Many published research papers and theses have used NS2, providing a valuable reference for us. The availability of extensive literature can help us validate our work against established benchmarks and facilitate comparisons with prior studies.

4.2.Performance Evaluation

4.2.1. Metrics for Performance Evaluation

The performance measures castoff in this simulation to improve the proposed solution network performance by increasing the net time and reducing the energy depletion over the existing work. Depending on the performance metrics, the recommended solution is compared to the SMAC and ESMAC. The following performance metrics are calculated for each round to compare the proposed solution to the existing work (Birla, 2018):

a. Average Energy Consumption

To obtain the total energy consumed by all nodes in the network, the energy consumption of each individual node is summed up. This includes the energy expended during sensing, processing, communication, and other activities performed by the nodes.

$$\text{Average Energy Consumption} = \frac{\text{Total Energy Consumed(n)}}{\text{Number of Node}} \text{----------------------------------} \; 4.1$$

b. Average Throughput

The average pace at which data is effectively carried from source nodes to destination nodes over a certain amount of time is referred to as a network's throughput. It displays the quantity of meaningful data that is sent through the network per unit of time. To determine the typical throughput:

$$\text{Average Throughput} = \frac{\text{Total Data Transmitted}}{\text{Time Take}} \text{---} \; 4.2$$

Measure Data Transmission: Keep track of the volume of data that is successfully transported over a given time period from source nodes to destination nodes. The number of packets or bytes that are transferred can be counted to accomplish this.

Determine Time Duration: Measure the duration of the time period over which you are calculating the average throughput.

c. Average Delay

The average time it takes a packet to go from its source node to its destination node is often referred to as a network's average latency. It is a statistic that measures the latency or delay that packets encounter while traveling over the network.

$$\text{Av\_Delay} = \frac{Total\ Delay}{Number\ of\ Packets} \text{--- } 4.3$$

Measure Packet Delays: Determine the delay that each packet encounters while traveling over the network. By timestamping packets at the source node and noting the arrival time at the destination node, this may be accomplished.

Collect Data: Measured delay values should be gathered for a significant number of packets. The needed number of packets is determined by the desired level of statistical certainty as well as the network's variability in latency.

Calculate Total Delay: Sum up the delay values of all the packets collected.

Determine Packet Count: Count the number of packets for which delay measurements were collected.

4.3. Result of Simulation

The comparison between the existing SMAC and ESMAC and the suggested EES-MAC in the presence of SLT and clusterization is shown in the simulation graphs below. Clustering can be used to address the effects of SLT in EES-MAC, which include lower average throughput and enhanced average energy saving. The proposed EES-MAC is contrasted with the SMAC and ESMAC MAC protocols that were already in use in the next section.

4.3.1. Simulation Configuration

The performance of the suggested work is compared to the current ESMAC and SMAC protocol using random topology. Two simulated scenarios were used for the evaluation: The first example involves clustering a network and increasing the duty cycle from 20 to 100. The second scenario involves employing SLT and network clustering to increase the number of nodes from 6 to 30. In both situations, we assess how well the suggested approach, EES-MAC with ESMAC and SMAC, performs in terms of energy use, average throughput, and average delay.

Table 4.1 describes the factors used by ESMAC to attain the available results. We were able to compare the consequences we got based on that to theirs. If we change their parameters, we may get a result that is above or below them. For example, if we convert routing protocol to DSR, we can get more results because, in cluster formation, DSR, DSDV, etc. are better than AODV, but we have difficulty comparing them, so we use those parameters as they are.

Simulation example of randomly distributed 30 nodes

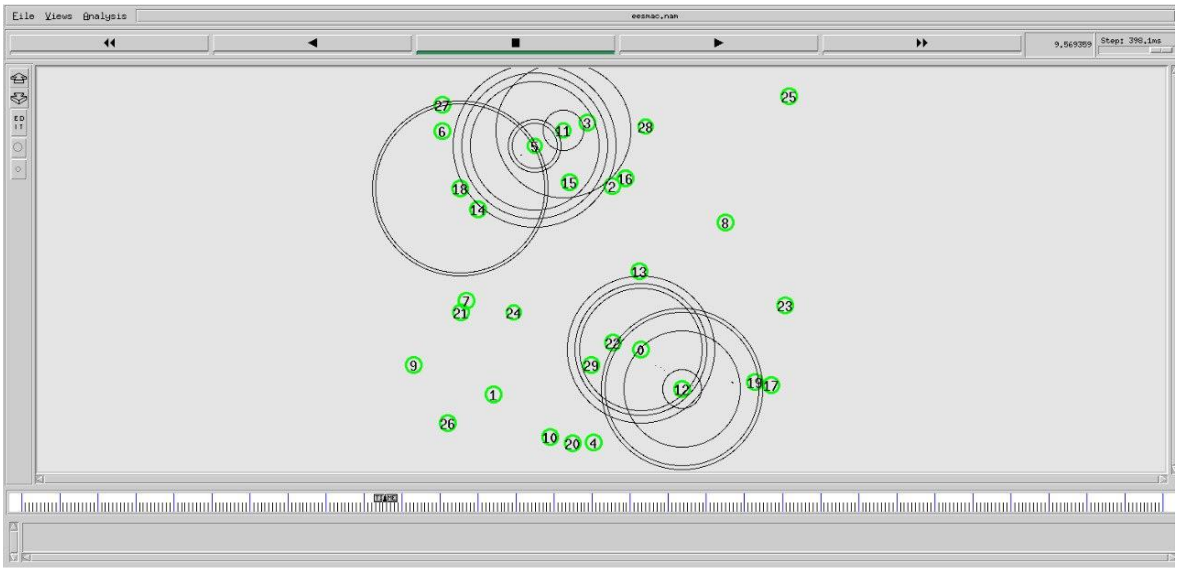


Figure 4.1 Sample random topology of 30 nodes

Table 4.1 Simulation configuration parameters

Parameters	Value
MAC protocol	Mac/SMAC
Topology configuration mode	Random
Routing protocol	AODV
Network area	801x502
Number of nodes	6, 10, 14, 16, 18, 20, 25, 30
Traffic	CBR
Agent	UDP
Initial energy	1000J
Idle power	1.0 watts
Transmission power	1.0 watts
Reception power	1.0 watts
Sleep power	0.001 watts
Transition power	0.2 watts
Transition time	0.005s
Duty cycle	20, 30, 40, 50, 60, 70, 80, 90, 100
Interface queue length	50
Simulation time	100s

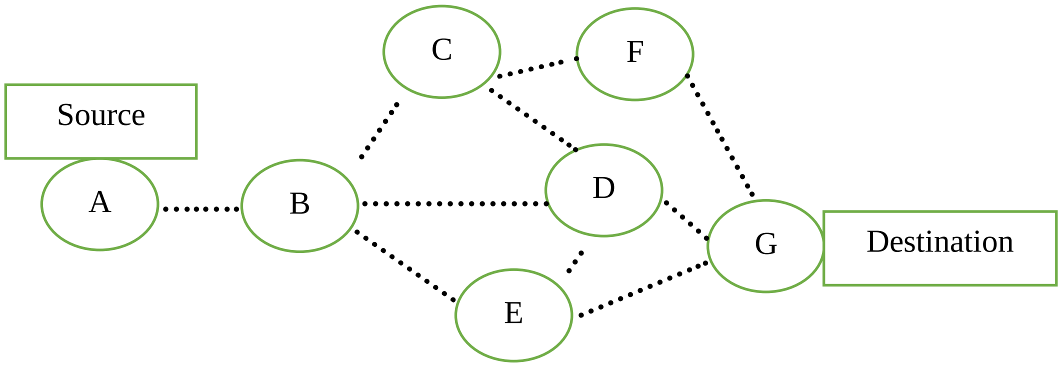


Figure 4.2 Sample diagram of node communication

S-MAC can generally only deliver a packet up to 2 hops every operational cycle since the hop following C (such as node F) is unlikely to have been awake to overhear the conversation from B to C. If Node D does not respond with a CTS after receiving an RTS from Node C, Node C will revert to sleep. The use of adaptive listening might possibly result in a significant increase in energy consumption due to the likelihood that multiple neighboring nodes may hear the RTS or CTS and awaken even if only one of them is the next-hop node. A node could also not be completely aware of the wireless medium's busy state since it doesn't wake up until a conversation being overheard is concluded. The packet may cause collisions at other nodes, for example, if the node begins sending any packet after missing an RTS or CTS of another data transmission in the vicinity. Based on this knowledge, ESMAC replicates SMAC in a multi-hop environment with an activated synchronization flag. In order to enhance throughput, they have used a number of node-based contention windows. But is not enough to improve throughput while saving energy consumption in the network. We improve it by using both a network size-based contention window and SLT supported by cluster head selection for groups created to perform activities of member nodes when they get sleep time. In almost all simulation scenarios, EES-MAC outperforms ESMAC in average throughput and average delay.

The performance assessment of the proposed EES-MAC with integrated SLT and group formation characteristics, as well as an adjustable contention window depending on network size, is covered in the next sections of this chapter.

4.4. Analysis of the Simulation's Outcome

4.4.1. The Comparison of Results in Energy Consumption

Performance comparison of energy consumption of EES-MAC, ESMAC, and SMAC with changes in duty-cycle. As the number of nodes rises, the EES-MAC protocol exhibits greater energy efficiency than ESMAC and SMAC. EES-MAC reduces energy usage by applying clusterization and modifying the duty cycle based on the remaining energy of nodes.

Figure 4.3 illustrates how, when employing both techniques, EES-MAC's energy usage is consistently lower than that of ESMAC and SMAC. The proposed work EES-MAC improved by 26.18% and 34.57% by changing duty cycle of nodes when compared to the existing works ESMAC and SMAC, respectively. The result tells us the capability of nodes to conserve more energy in different application like forest fire detection, healthy, military, environmental monitoring and etc. Sleep time of nodes increased which result energy saving and there is also CH that perform activity of other nodes in his group which enhance throughput of the network than both ESMAC and SMAC. The improvement in network lifespan is caused by the decrement in energy consumption rate and other elements incorporated into the fitness function used in the proposed study.

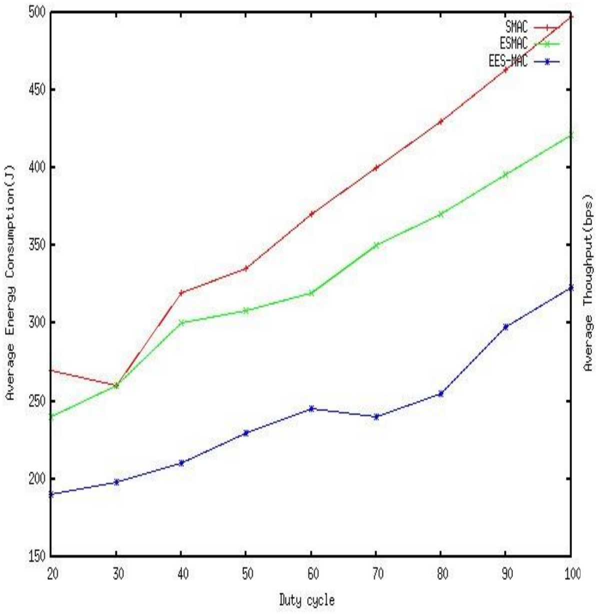


Figure 4.3 Average energy consumption with change in duty-cycle.

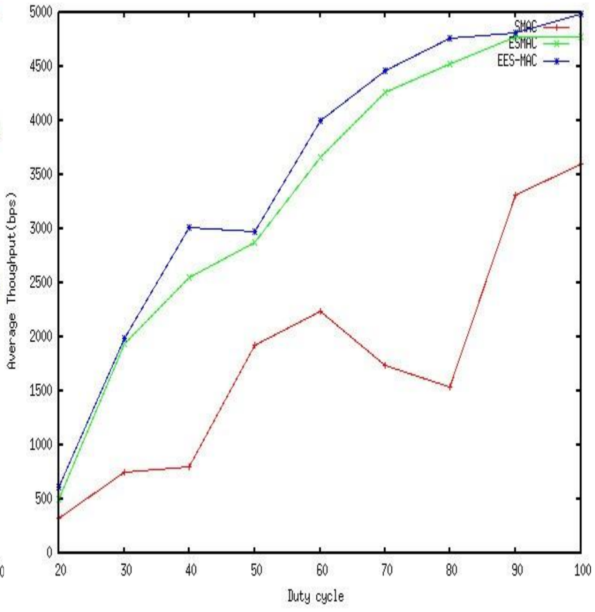


Figure 4.4 Average throughput as duty-cycle changes.

4.4.2. The Comparison of Results in Terms of Throughput

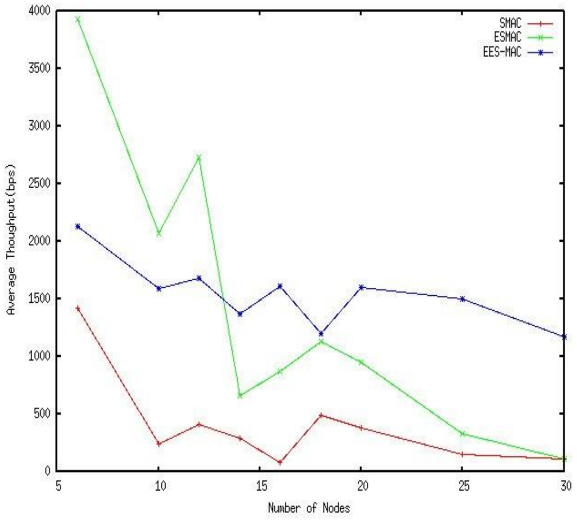
For some applications, great throughput is necessary while yet conserving SN energy. By creating a cluster with a cluster head that carries out the functions of member nodes, such as notifying other nodes when it is available by sending its updated synchronization, receiving acknowledgement from group members, and sending it when it is in listen mode for improving throughput, we add extra features to ESMAC for such applications.

The proposed work, EES-MAC, achieves better performance in throughput as the duty cycle increases from 20 to 100. As demonstrated in the Figure 4.4 the proposed work EES-MAC improved by 5.89% and 95.28% by changing the duty cycle when compared to the existing works ESMAC and SMAC, respectively. This is because the tasks that should have been done by the nodes are done by the CH so that the performance of the network doesn’t degrade while they are sleeping. This one keeps the throughput from going decreased.

The performance improvement in EES-MAC is attributed to the tasks being done by the Cluster Heads (CHs) instead of the individual nodes. When the nodes are in sleep mode, the CHs take over those tasks, ensuring that the network's performance does not degrade. This approach of offloading tasks to CHs helps maintain or even increase the network throughput, instead of allowing it to decrease.

As the duty cycle rises from 6 to 30, the proposed task, EES-MAC, produces a higher throughput. As shown in Figure 4.5, the suggested work EES-MAC outperformed the current works ESMAC and SMAC by 27.34% and 74.20%, respectively, by altering the number of nodes. Performance does not drop as the number of nodes rises since more groups are more likely to exist. This has prevented EESMAC's performance from being subpar compared to the others.

These results suggest that the EES-MAC algorithm has the potential to be advantageous in situations that require both fast data transmission and efficient throughput. Various applications in WSN such as environmental monitoring, surveillance systems, industrial automation, precision agriculture, and smart city infrastructure can benefit from the enhancements provided by EES-MAC. By enhancing data throughput and utilizing the advantages of node grouping, EES-MAC can significantly improve performance, resulting in more dependable and efficient data collection, monitoring, and control in WSN applications.



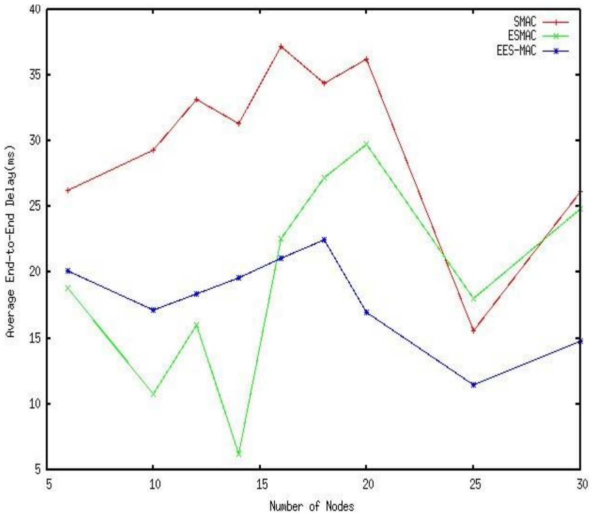


Figure 4.5 Average throughput of 30 nodes

Figure 4.6 Changes in average delay with node number

4.4.3. Result Comparison in Delay

Since SNs send data across numerous hops, each hop's latency is increased since it takes longer to wait for the channel before sending data. But as figure 4.6 shows, delay is improved in EES-MAC since we use clusterization, in which cluster heads perform the activity of other members' nodes under sleep.

As shown in Figure 4.6, when we adjust the number of nodes and duty cycle, the simulation shows that the average delay of the proposed EES-MAC is less than the present ESMAC and SMAC in both circumstances. Since the suggested system includes a motivational mechanism that switches a node's wake-up mode to sleep mode when it doesn't conduct any activity, as well as by altering cluster heads that handle tasks for nodes in sleep mode in his group, the proposed system has these features.

As a case in point, the current SMAC and ESMAC's average latency was 8.99 msec and 6.37 msec, respectively, and it was reduced to 5.70 msec after SLT and clusterization were applied. The average delay of the SMAC is reduced by 36.52%, and that of the ESMAC is reduced by 10.52% when node numbers change.

EES-MAC exhibits an improved average delay compared to ESMAC and SMAC. The use of clusterization in EES-MAC enables cluster heads to perform activities on behalf of sleeping member nodes, reducing the time spent listening to the channel at each hop and minimizing delay.

The reduction in average delay signifies an improvement in the efficiency and responsiveness of the network. Lower average delay values indicate faster packet delivery, reduced latency, and potentially enhanced real-time data transmission capabilities. This can be particularly beneficial in applications where timely data collection, event reporting, or control actions are crucial, such as environmental monitoring, surveillance, healthcare, and industrial automation systems. By reducing the average delay through the implementation of SLT and clusterization, EE-SMAC protocols become more suitable for these applications, enabling improved performance and reliability in WSN.

Table 4.2 Results of simulation when the number of nodes is changed

EES-MAC			ESMAC		SMAC	
Nodes Numbers	Throughput	Delay	Throughput	Delay	Throughput	Delay
6	2130.47	20.12	2025.47	18.78	1414.50	26.25
10	1590.12	17.12	2073.32	10.75	240.46	29.35
12	1678.09	18.36	2730.47	16.03	412.22	33.19
14	1365.56	19.65	658.72	16.23	285.62	31.31
16	1612.02	21.08	869.61	22.56	80.34	37.19
18	1199.17	22.49	1128.16	27.20	491.58	34.43
20	1600.01	17.01	945.76	29.78	382.21	36.20
25	1496.76	18.50	326.96	24.91	150.11	15.56
30	1173.70	16.81	114.52	24.84	114.70	26.13

Due to the increased listening time of nodes, as we increase the number of nodes and their duty cycles, there will be more energy consumption in SMAC and ESMAC when compared to EES-MAC. The cluster head of those members performs all of the activities of nodes that are in sleep mode in EES-MAC, where the listen time of SNs is defined by their energy level. If there is no work, a node should not idle and wait for the duty cycle to complete its time. As the sleep time of the SNs increases, energy consumption decreases in the EES-MAC compared to the existing ESMAC and SMAC.

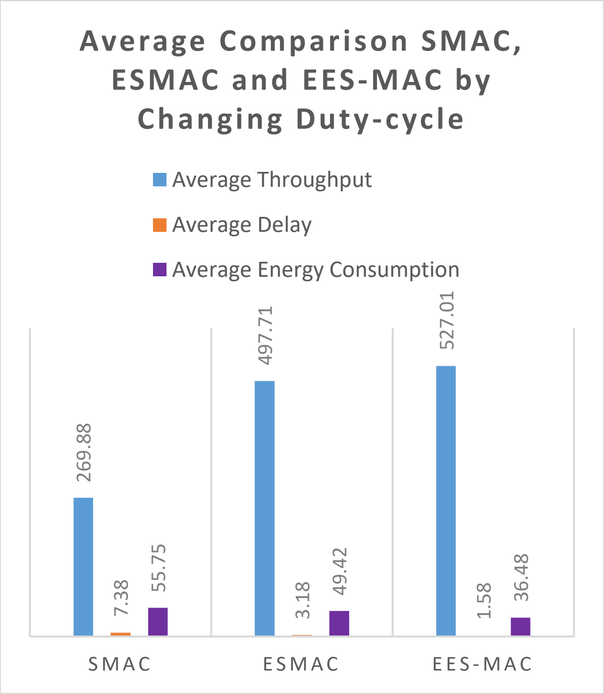
EES-MAC outperforms ESMAC and SMAC in terms of throughput, especially as the duty cycle increases. The utilization of SLT adjustment and clusterization in EES-MAC allows for SLT and improves the overall throughput of the network. While clusterization aids in overcoming the throughput reduction brought on by SLT, SLT conserves energy by lengthening the sleep period of nodes based on their remaining energy.

Table 4.3 Simulation result by change in duty cycle

EES-MAC				ESMAC			SMAC		
Duty Cycle	Throu.	Delay	Ener. Cons.	Throu.	Delay	Ener. Cons.	Throu.	Delay	Ener. Cons.
20	610.31	30.62	190	505.88	78.86	240	319.77	127.49	270
30	1990.72	21.15	198	1936.65	26.06	260	746.57	93.77	260
40	3016.56	10.51	210	2549.38	20.52	300	796.39	85.18	320
50	2978.36	5.50	230	2870.77	9.90	308	1922.67	26.11	335
60	3997.12	7.56	245	3662.40	12.28	320	2232.09	21.34	370
70	4463.03	4.70	240	4267.18	14.28	350	1732.35	23.12	400
80	4761.90	5.00	255	4518.46	9.58	370	1541.07	41.79	430
90	4811.43	6.45	298	4778.95	9.81	396	3307.35	13.20	463
100	4991.12	3.21	323	4773.12	9.38	421	3594.27	10.94	497

The main difficulty in WSN design is to achieve energy economy while transmitting sensed data with a short latency and increased throughput. We employ clusterization and SLT to lower the average energy consumption of EES-MAC in comparison to SMAC and ESMAC. EES-MAC changes the duty cycle to 0.80, 0.60, and 0.30 of the initial duty cycle as the node's residual energy falls below 0.80, 0.60, and 0.30 of the starting energy. Duty cycle, which measures the ratio of listening time to sleep time, can be reduced to give nodes more time to sleep. We can improve the functionality of SMAC and the ESMAC protocol thanks to the SLT adjustment and clusterization adjustment techniques employed in EES-MAC.

As the number of nodes rises, the EES-MAC protocol exhibits greater energy efficiency than ESMAC and SMAC. As shown in Figure 4.7, the average throughput rose in the EES-MAC to 527.01 from 269.88 and 497.71, the average latency was reduced from 7.38 to 3.18 to 1.58, and the average energy consumption was reduced from 55.75 to 49.42 in the SMAC and ESMAC, respectively. By changing the duty cycle, EES-MAC achieves better average energy consumption.



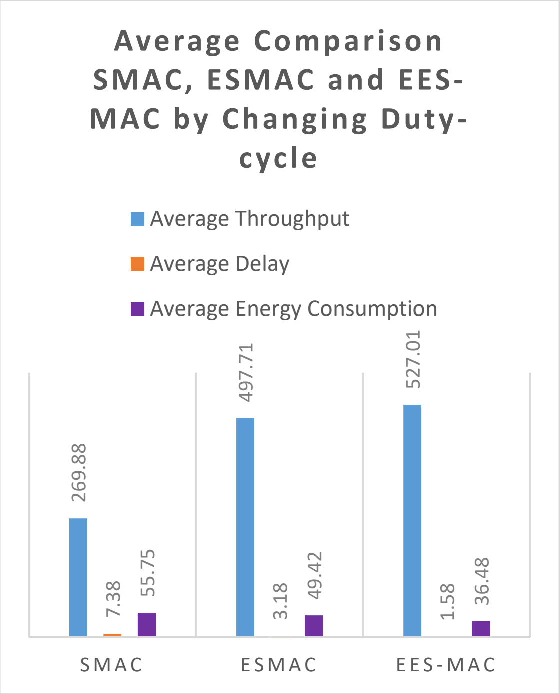


Figure 4.7 Average comparison by changing duty cycle

Figure 4.8 Average comparison by changing number of node

In EES-MAC, average throughput increased to 461.53 from 119.06 and 362.43, and average delay decreased to 5.7 from 8.99 and 6.37 in SMAC and ESMAC, respectively. By changing the number of nodes, EES-MAC achieves lower average energy consumption as we observe from Figure 4.8.

4.5.Discussion

The proposed EES-MAC and the two existing works, namely SMAC and ESMAC, were simulated in the NS2 simulation tool using a comparable number of nodes and the identical evaluation settings in order to evaluate the results based on the three key metrics. In comparison to the two current systems, the suggested EES-MAC performs better in terms of energy usage, throughput, and latency. The suggested EES-MAC provides a relative improvement over the current SMAC and ESMAC, according to Tables 4.2 and 4.3.

Following a discussion of the findings and the performance assessment measures presented, the research questions listed in Section 1.4 that our study sought to examine and answer will be as follows:

RQ1: What approaches can be taken to develop a new algorithm that enhances the performance of ESMAC? Several approaches have bene taken to develop a new algorithm that enhances the performance of ESMAC. Here are a few possible approaches:

Optimized Duty Cycling which explore different duty cycling schemes and optimization techniques to improve energy efficiency. This is involved by incorporating adaptive duty cycling mechanisms, or dynamically adjusting sleep/wake schedules to minimize energy consumption while maintaining sufficient connectivity. SLT intend to increase energy conservation and throughput, respectively. By making nodes in order they sleep those under wakeup mode while they are not performing any activity. SLT decreases the listening time of nodes to decrease idle time in networks by decreasing the duty cycle of the nodes. As a result of the SLT throughput of the network has become low. Clusterization optimize the cluster formation mechanisms to enhance scalability and data aggregation in ESMAC. We have developed more efficient cluster head election algorithms based on inter-cluster distance and remaining energy of SNs.

We utilize simulation tools or testbeds to evaluate the performance of the new algorithm in various scenarios. Conduct thorough performance analysis, comparing the proposed algorithm against existing MAC protocols, and considering metrics such as energy efficiency, network throughput, and delay.

By considering these approaches, we have developed that a new algorithm called SLT and clusterization that enhances the performance of ESMAC, contributing to improved energy efficiency, scalability, and overall performance in wireless sensor networks.

RQ2: What is the procedure for simulating EES-MAC using SLT and the clusterization algorithm?

We define the network topology by specifying the number of sensor nodes, their locations, and the communication range between nodes. This step establishes the foundation for simulating the WSN. Then we implement the EES-MAC protocol according to its specifications. This involves incorporating the SLT mechanism and the clusterization algorithm into the MAC layer of the simulation framework.

We simulated our proposed new system using various tools. We have simulated an algorithm designed using NS2 in OTCL and C++ code. We have added the parameters it requires, such as node number, duty cycle, routing protocol, and so on, which are similar to those of the existing MAC protocol. It includes a network animator for output display as well as a trace file. We chose NS2 because it is open, simple, and suitable for simulating WSN.

RQ3: How to evaluate the performance of EES-MAC with the existing ESMAC and SMAC?

To evaluate the performance of the EES-MAC protocol in comparison to existing protocols such as ESMAC and SMAC, we had to consider the following evaluation metrics:

Throughput: Measure the amount of data successfully transmitted over a given period of time. Compare the throughput achieved by EES-MAC with that of ESMAC and SMAC under similar conditions.

Average Delay: Determine the average time it takes for packets to traverse the network. Compare the average delay of EES-MAC with ESMAC and SMAC to assess the efficiency of packet transmission.

Energy Efficiency: Evaluate the energy consumption of each protocol. Compare the energy efficiency of EES-MAC with ESMAC and SMAC to determine which protocol utilizes energy more effectively.

SMAC, ESMAC, and EES-MAC simulation data were used in the performance evaluation. Using three evaluation criteria as mentioned in Sec. 4.2.1, we next evaluated the performance of the newly constructed EES-MAC solution in comparison to the pre-existing SMAC and ESMAC. The following Table 4.4 illustrates how much such performance indicators are enhanced by the proposed EES-MAC. The average throughput of the net was increased in the proposed solution by 5.89% over ESMAC and 95.28% over SMAC. The average delay was decreased in the proposed solution by 50.31% over ESMAC and 78.6% over ESMAC. The Average energy consumption was decreased in the proposed solution by 26.18% over ESMAC and 34.57% over SMAC.

Table 4.4 Performance assessment of EES-MAC

Evaluation metrics	Improvement when changing duty cycle	
	ESMAC	SMAC
Average Throughput	↑ 5.89%	↑ 95.28%
Average Delay	↓ 50.31%	↓ 78.60%
Average Energy Consumption	↓ 26.18%	↓ 34.57%

Here, the energy consumed is minimized by making the idle nodes sleep. We often see them being quiet without work because they have time, which is a waste of energy. Another thing is to avoid performance decreasing when nodes are asleep. If there is work after they are asleep, we have created a group and selected a leader for that group to do the work of nodes that are asleep while they have time.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1. Conclusion

WSNs have developed into a powerful technology with many uses. They are composed of tiny, low-cost, energy-restricted SNs that collaborate to wirelessly collect and deliver data. Scalability, flexibility, and the ability to remotely adjust physical settings are just a few advantages that WSNs provide. WSNs are currently being used in a number of applications, from ecological monitoring to housing monitoring, by putting them in a specific application.

WSNs must additionally contend with issues including energy, security, and dependability. Since SNs are frequently powered by energy harvesters with limited capacity, energy efficiency is a crucial issue. Effective power management strategies, including node sleep scheduling, data aggregation, and others, are necessary to extend the network's life. This thesis takes into consideration MAC mechanisms as a significant strategy to enable smart use of restricted sensor node resources since wireless sensor nodes are resource-constrained nodes that need to be. In WSNs, MAC protocols fall into three broad categories. These protocols are hybrid, contention-based, and contention-free. The EES-MAC protocol is suggested in this thesis as a means of addressing the well-known Contention-based ESMAC protocol's weaknesses in terms of different parameters.

In this thesis, an EES-MAC is proposed to enhance network lifetime while simultaneously increasing throughput and lowering the energy consumption of SNs. We design nodes to have SLTs, the length of which is dependent on the amount of energy that each node has left; nodes with more energy have longer listening times, and those with less energy have SLT. Additionally, we create the concept of clusterization, which addresses the issue brought up by the use of SLT, which reduces network throughput. Clusterization depends on how far apart nodes are from one another, and CH choice is dependent on a node's lingering energy. By using those mechanisms, the developed protocol generally increased the performance of the SMAC and ESMAC protocols that were already developed.

5.2. Contribution

The key contribution of this study is to save energy consumed during the listening time of nodes when there is no activity performed for different reasons such as ideal listening. By increasing the sleeping time of nodes we save energy through the SLT and clusterization we increase the throughput recital of the network that degraded as a result of using SLT.

Usually the nodes consume energy quietly even if they are idle, as long as they don't finish the time allotted to them in the duty cycle. In this one, we have made the nodes sleep if there is no work while they have listening time. And maybe since node’s asleep, we've had the CH of the team do it on his behalf if there's work to be done. In this way, we have been able to save energy. In general, there is no work left due to a lack of time. And there won't be any wasted energy by getting longer. The middling throughput of the net was increased in the proposed solution by 5.89% over ESMAC and 95.28% over SMAC. The average delay was decreased in the proposed solution by 50.31% over ESMAC and 78.6% over ESMAC. The Average energy consumption was decreased in the proposed solution by 26.18% over ESMAC and 34.57% over SMAC.

The energy efficiency achieved through SLT and clusterization contributes to prolonging the network lifetime. By optimizing energy consumption and workload distribution, the proposed MAC protocol can extend the operational time of the network before nodes start depleting their energy reserves. This is particularly valuable in applications where frequent battery replacement or recharging is not feasible or practical.

The designed energy-efficient MAC protocol using SLT and clusterization has the potential to benefit a wide range of WSN applications. Whether it's environmental monitoring, surveillance, healthcare, smart agriculture, or industrial automation, the protocol's energy-saving capabilities and improved performance can make WSNs more viable and reliable for various real-world applications, like in the military for detection of enemy movement and intrusion detection; in the environment for detection of forest fires and prevention of natural disasters; and researchers can use this work as an input for providing better optimization.

5.3. Future Works

The primary objective of the designed protocol in this study is to enhance the network's lifespan by creating an energy-efficient MAC protocol. However, there are several unresolved issues that remain outside the scope of this research, making it crucial for future researchers to consider this study as a foundation for investigating ways to improve the lifetime of SNs in a broader context.

The proposed work was done using both the SLT and clusterization algorithms. However, our clusterization is based on only the inter-cluster distance and their residual energy. But more energy can be saved if clusters can be formed in addition based on node density and the amount of energy nodes consume.

Special Acknowledgement

This research was sponsored by Adama Science and Technology University with a grant number

ASTU/SM-R/817/23 Adama, Ethiopia

REFERENCES

Abrach, H., Bhatti, S., Carlson, J., Dai, H., Rose, J., Sheth, A., Shucker, B., Deng, J., & Han, R. (2003). MANTIS: System Support for MultimodAl NeTworks of In-situ Sensors. *Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications, WSNA 2003*, 50–59.

Akhmetshina, E., Gburzynski, P., & Vizeacoumar, F. (2003). PicOS: A Tiny Operating System for Extremely Small Embedded Platforms. *Proceedings of the International Conference on Embedded Systems and Applications*, *January 2003*, 116–122.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, *38*(4), 393–422. https://doi.org/10.1016/S1389-1286(01)00302-4

Alemdar, A., & Ibnkahla, M. (2007). Wireless sensor networks: Applications and challenges. *2007 9th International Symposium on Signal Processing and Its Applications, ISSPA 2007, Proceedings*. https://doi.org/10.1109/ISSPA.2007.4555630

Alfayez, F., Hammoudeh, M., & Abuarqoub, A. (2015). A Survey on MAC Protocols for Duty-cycled Wireless Sensor Networks. *Procedia Computer Science*, *73*(Awict), 482–489. https://doi.org/10.1016/j.procs.2015.12.034

Arshad, S., Al-Sadi, A. A., & Barnawi, A. Y. (2013). Z-MAC: Performance Evaluation and Enhancements. *EUSPN/ICTH*, *21*, 485–490. https://doi.org/https://doi.org/10.1016/j.procs.2013.09.066.

Bakare, B. I., & Enoch, J. D. (2019). A review of simulation techniques for some wireless communication system. *International Journal of Electronics Communication and Computer Engineering*, *10*(2), 60–70.

Birla, J. (2018). Performance Metrics in Wireless Sensor Network. *International Journal of Engineering Research & Technology*, *3*(10), 1–4. www.ijert.org

Buckner, C. A., Lafrenie, R. M., Leong, Y. C., Bee, P. C., Chin, E., Teh, A. K. H., Picco, S., Tonelli, F., & Mathijssen, R. H. J. (2016). WSNs – Applications. *Intech*, *11*(tourism), 6–7. https://www.academia.edu/89023723/Wireless_Sensor_Networks_Applications

Daas, M. S., Chikhi, S., & Bourennane, E. B. (2021). A dynamic multi-sink routing protocol for static and mobile self-organizing wireless networks: A routing protocol for Internet of Things. *Ad Hoc Networks*, *117*(March), 102495. https://doi.org/10.1016/j.adhoc.2021.102495

Daesuk, Y., Susung, P., & Seungsink, C. (2008). Dynamic S-MAC protocol for wireless sensor networks based on network traffic states. *IEEE*.

Dam, T. van, & Koen, L. (2003). An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Association for Computing Machinery*, *1581137079*, 171–180. https://doi.org/doi = {10.1145/958491.958512},

Datta, U., Mukherjee, A., Sahu, P. K., & Kundu, S. (2013). Resource utilization of multi-hop CDMA wireless sensor networks with efficient forwarding protocols. *Procedia Engineering*, *64*, 46–55. https://doi.org/10.1016/j.proeng.2013.09.075

Dattatraya, K. N., & Rao, K. R. (2022). Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN. *Journal of King Saud University - Computer and Information Sciences*, *34*(3), 716–726. https://doi.org/10.1016/j.jksuci.2019.04.003

Dias, G. M., Margi, C. B., De Oliveira, F. C. P., & Bellalta, B. (2019). Cloud-Empowered, Self-Managing Wireless Sensor Networks: Interconnecting Management Operations at the Application Layer. *IEEE Consumer Electronics Magazine*, *8*(1), 55–60. https://doi.org/10.1109/MCE.2018.2868110

Engelmann, L., Humphrey, C., & Lynteris, C. (2022). Working with diagrams. In *Working With Diagrams*. https://doi.org/10.3167/9781800735583

Farooq, M. O., & Kunz, T. (2011). Operating systems for wireless sensor networks: A survey. *Sensors*, *11*(6), 5900–5930. https://doi.org/10.3390/s110605900

Fong, D. Y. (2017). Design Model and Deployment Fashion of Wireless Sensor Networks. *Internet of Things and Data Analytics Handbook*, 197–213. https://doi.org/10.1002/9781119173601.ch12

Gherardi, F., & Aquiloni, L. (2011). A Survey on Sensor Networks. *Crustaceana Monographs*, *15*(August), 213–223. https://doi.org/10.1163/ej.9789004174252.i-354.145

He, X., Liu, S., Yang, G., & Xiong, N. (2018). Achieving efficient data collection in heterogeneous sensing WSNs. *IEEE Access*, *6*(c), 63187–63199.

https://doi.org/10.1109/ACCESS.2018.2876552

Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., & Ganesan, D. (2001). *Building efficient wireless sensor networks with low-level naming*. 146–159. https://doi.org/10.1145/502034.502049

Henna, S. (2012). SA-RI-MAC: Sender-assisted receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 81 LNICST*, 120–135. https://doi.org/10.1007/978-3-642-29479-2_10

Hoblos, G., Staroswiecki, M., & Aitouche, A. (2000). Optimal design of fault tolerant sensor networks. *IEEE Conference on Control Applications - Proceedings*, *1*, 467–472. https://doi.org/10.1109/cca.2000.897468

Islam, M., Kumar, A., & Hossain, A. (2019). Study of Wireless Sensor Network. *International Journal of Sensors and Sensor Networks*, *7*(1), 9. https://doi.org/10.11648/j.ijssn.20190701.12

Iyekekpolo, U. B., Idachaba, F. E., Popoola, S. I., Atayero, A. A., & Mensah, F. (2018). Wireless sensor networks: Architecture, applications, and challenges. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, *2018*(SEP), 1212–1220.

Jain, V., Agarwal, S., & Goswami, K. (2014). Dynamic multilevel priority packet scheduling design for WSN. *2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014*, 86–90. https://doi.org/10.1109/ICSPCT.2014.6884950

Jandaeng, C., Suntiamontut, W., & Elz, N. (2011). PSA: The Packet Scheduling Algorithm for Wireless Sensor Networks. *International Journal on Applications of Graph Theory In Wireless Ad Hoc Networks And Sensor Networks*, *3*(3), 1–12. https://doi.org/10.5121/jgraphoc.2011.3301

Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M., & Ismail, M. (2017). Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors (Basel, Switzerland)*, *17*(8). https://doi.org/10.3390/s17081781

Jondhale, S. R., Maheswar, R., & Lloret Mauri, J. (2022). *Received signal strength based target localization and tracking using wireless sensor networks*.

Kumar, R., & Gangwar, M. (2023). Improved BEST-MAC protocol for WSN using optimal cluster head selection. *International Journal of Information Technology (Singapore)*, *15*(2), 859–875. https://doi.org/10.1007/s41870-019-00385-9

Labs, S. (2004). The Evolution of Wireless Sensor Networks. In *The Evolution of Wireless Sensor Networks* (pp. 1–5). http://www.silabs.com/Support Documents/TechnicalDocs/evolution-of-wireless-sensor-networks.pdf

Lin, D., Wang, Q., Min, W., Xu, J., & Zhang, Z. (2020). A Survey on Energy-Efficient Strategies in Static Wireless Sensor Networks. *ACM Trans. Sen. Netw.*, *17*(1), 1–48. https://doi.org/10.1145/3414315

Nwankwo, W., & Ukhurebor, K. E. (2019). Investigating the performance of point to multipoint microwave connectivity across undulating landscape during rainfall. *Journal of the Nigerian Society of Physical Sciences*, *1*(3), 103–115. https://doi.org/10.46481/jnsps.2019.16

Obaidat, M. S., & Misra, S. (2014). Wireless sensor network applications: overview and case studies. *In Principles of Wireless Sensor Networks, Cambridge: Cambridge University Press.*, 30–47. https://doi.org/10.1017/cbo9781139030960.004

Pang, B., Shi, H., & Li, Y. (2012). An Energy-Efficient MAC Protocol for Wireless Sensor. *IEEE*, 163–164.

Rabaey, J. M., Ammer, M. J., Da Silva, J. L., Patel, D., & Roundy, S. (2000). PicoRadio supports ad hoc ultra-low power wireless networking. *Computer*, *33*(7), 42–48. https://doi.org/10.1109/2.869369

Rastogi, T. (2017). *A Review on Routing Protocols of Wireless Sensor Network*. *4*(04).

Razaque, A., Almiani, M., Khan, M. J., Kakenuly, M. N., & Jararweh, Y. (2018). Energy efficient medium access control protocol for wireless sensor networks. *Proceedings of 2018 6th International Renewable and Sustainable Energy Conference, IRSEC 2018*, 1–6. https://doi.org/10.1109/IRSEC.2018.8703030

Sahoo, S. K., & Nayak, A. K. (2019). A research on energy efficient multiple query optimization in WSN. *International Journal of Recent Technology and Engineering*, *8*(2 Special Issue 11), 2158–2162. https://doi.org/10.35940/ijrte.B1227.0982S1119

Sakya, G., & Sharma, V. (2019). ADMC-MAC: Energy efficient adaptive MAC protocol for mission critical applications in WSN. *Sustainable Computing: Informatics and Systems*, *23*, 21–28. https://doi.org/10.1016/j.suscom.2019.05.001

Shafiullah, G., M., Azad, S., Ali, & A. (2013). Energy-efficient wireless MAC protocols for railway mon_itoring applications. *IEEE Trans. Intell. Trans. Syst. 14(2)*, 649–659.

Shahraki, A., Taherkordi, A., Haugen, Ø., & Eliassen, F. (2020). Clustering objectives in wireless sensor networks: A survey and research direction analysis. *Computer Networks*, *180*(June). https://doi.org/10.1016/j.comnet.2020.107376

Shaiful, A. C., & Mohammed, T. I. (2010). Performance analysis of CSMA and IEEE 802.11 in Wireless Sensor Networks using GloMoSim and IEEE. *International Journal of Computer Science and Security*, 81.

Shih, E., Cho, S. H., Ickes, N., Min, R., Sinha, A., Wang, A., & Chandrakasan, A. (2001). Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 272–286. https://doi.org/10.1145/381677.381703

Silva, F., Heidemann, J., Govindan, R., & Estrin, D. (2004). Directed diffusion. *Distributed Sensor Networks*, 573–596. https://doi.org/10.1201/9780203487068.ch29

Sinha, A., & Chandrakasan, A. P. (2001). Operating system and algorithmic techniques for energy scalable wireless sensor networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1987*, 199–209. https://doi.org/10.1007/3-540-44498-x_16

Sohraby, K., & Al., E. (2007). *WIRELESS SENSOR NETWORKS Technology, Protocols, and Applications*. John Wiley & Sons, Inc. https://doi.org/https://doi.org/10.1002/9780470112762.ch3

Song Feiyan, Y. B. and Y. Z. (2016). A Novel Sleeping Scheduling Method for Wireless Sensor Networks Based on Data Fusion. *International Journal of Future Generation Communication and Networking*, 155–162.

Su, W., Akan, O. B., & Cayirci, E. (2004). Chapter 2 COMMUNICATION PROTOCOLS FOR SENSOR. *Networks*, 21–50. https://doi.org/10.1007/978-1-4020-7884-2_2

Sun, Y., Du, S., Johnson, D. B., & Gurewitz, O. (2008). DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. *Proceedings of* *the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 53–62. https://doi.org/10.1145/1374618.1374627

Teshome, D. (2018). Enhanced Sensor Medium Access Control Protocol for Wireless Sensor Network. *Unpublished*. http://10.140.5.162//handle/123456789/1798

Tosi, J., Taffoni, F., Santacatterina, M., Sannino, R., & Formica, D. (2019). Throughput Analysis of BLE Sensor Network for Motion Tracking of Human Movements. *IEEE Sensors Journal*, *19*(1), 370–377. https://doi.org/10.1109/JSEN.2018.2877102

Wang, C., Chen, Y., & Hou, Y. (2013). The analysis and improvement of SMAC protocol for wireless sensor networks. *Proceedings - IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2013*, *4122009*, 437–441. https://doi.org/10.1109/MSN.2013.15

Wei, Y., John, H., & Deborah, E. (2002). An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *IEEE*.

Weixia, Z., Wang, E., Zhou, Z., & Li, W. (2012). A Contention Window Adaptive MAC Protocol for Wireless Sensor Networks. *7th International ICST Conference on Communications and Networking in China (CHINACOM)*.

Yalçın, S., & Erdem, E. (2020). A mobile sink path planning for wireless sensor networks based on priority-ordered dependent nonparametric trees. *International Journal of Communication Systems*, *33*(12), 1–19. https://doi.org/10.1002/dac.4449

Ye, W., Heidemann, J., Estrin, D., & An. (2002). energy-efficient MAC protocol for wireless sensor networks. *In: Proc. of the 21st Conf. of the IEEE Computer and Communications Societies (INFOCOM). Volume*, *3*, 1567–1576.

Zhang, J., Li, W., Cui, D., Zhao, X., & Yin, Z. (2009). The NS2-based simulation and research on wireless sensor network route protocol. *Proceedings - 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009*, 4–7. https://doi.org/10.1109/WICOM.2009.5302699

Zou, W., Wang, E., Zhou, Z., Li, W., & Ye, Y. (2012). A contention window adaptive MAC protocol for Wireless Sensor Networks. *7th International Conference on Communications and Networking in China*, 566-570.

APPENDIXS

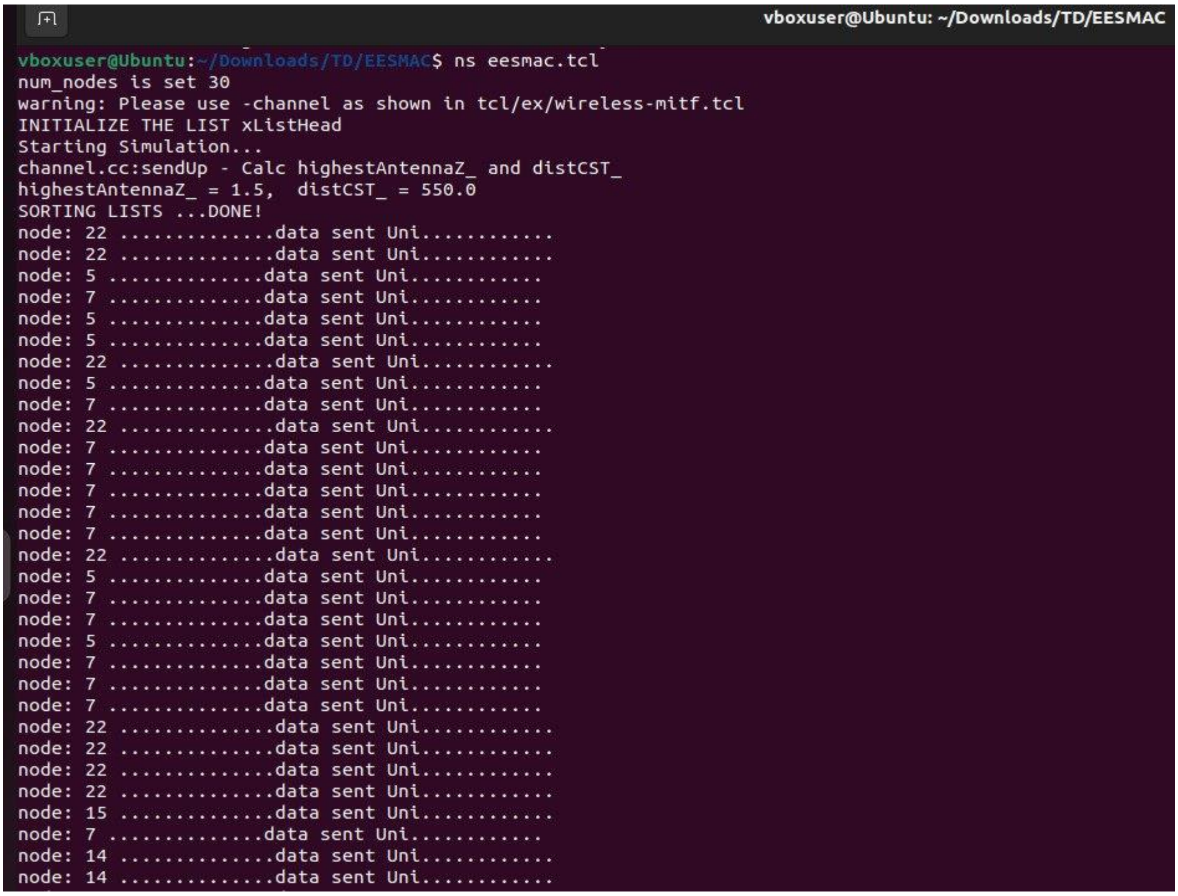
Appendix A: Simulation parameters (tcl script)

```
#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     30                         ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      1001                       ;# X dimension of topography
set val(y)      656                        ;# Y dimension of topography
set val(stop)   30.0                       ;# time of simulation end
#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]
#Setup topography object
```

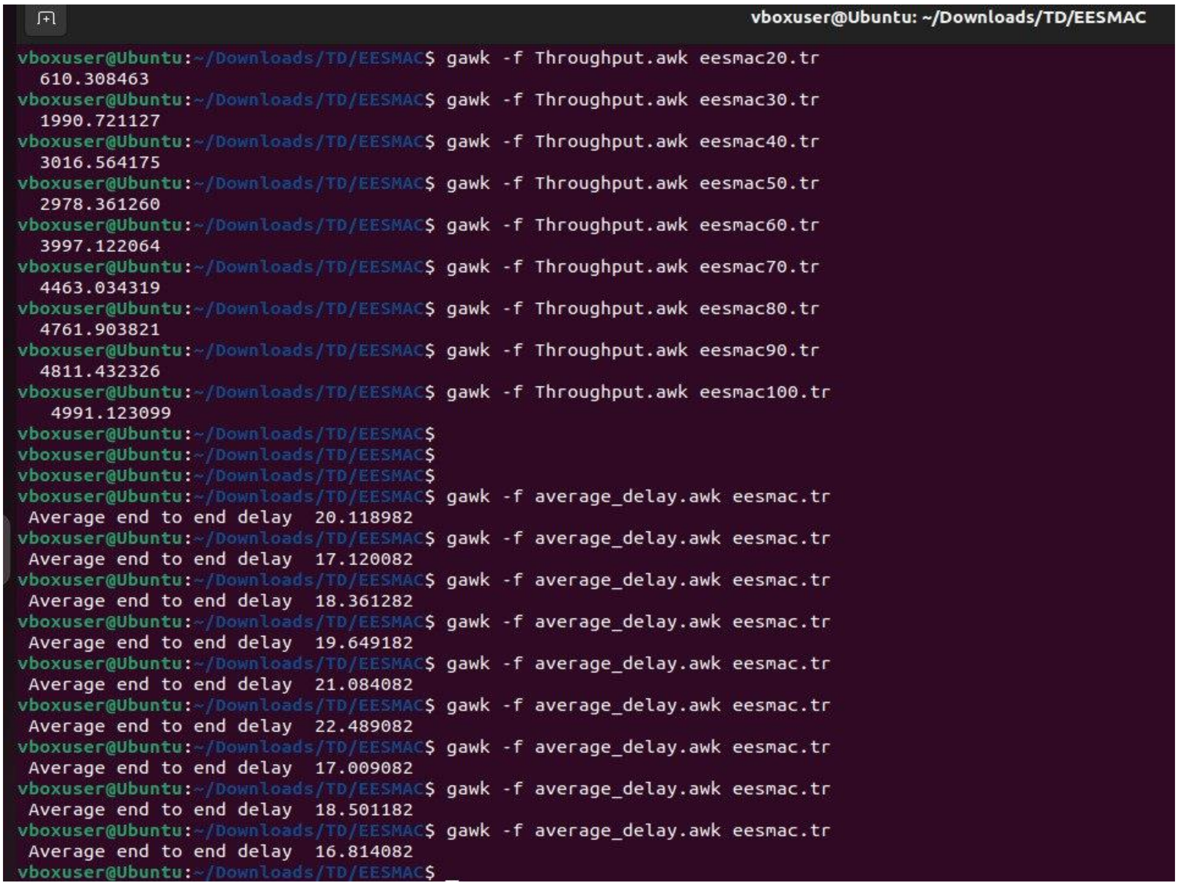
```
set topo       [new Topography]

$topo load_flatgrid $val(x) $val(y)

#===================================

#GOD Creation - General Operations Director

#===================================

create-god $val(nn)

set channel1 [new $val(chan)]

set channel2 [new $val(chan)]

set channel3 [new $val(chan)]

#===================================

#Open the NS trace file

#===================================

set tracefile [open eesmac.tr w]

$ns trace-all $tracefile

#Open the NAM trace file

set namfile [open eesmac.nam w]

$ns namtrace-all $namfile

$ns namtrace-all-wireless $namfile $val(x) $val(y)

$ns use-newtrace

set chan [new $val(chan)];#Create wireless channel

#===================================

#     Mobile node parameter setup

#===================================

$ns node-config -adhocRouting  $val(rp) \
```

```
          -llType        $val(ll) \
          -macType       $val(mac) \
          -ifqType       $val(ifq) \
          -ifqLen        $val(ifqlen) \
          -antType       $val(ant) \
          -propType      $val(prop) \
          -phyType       $val(netif) \
          -topoInstance  $topo \
          -energyModel "EnergyModel" \
          -initialEnergy 1000.0 \
-txPower 1.0 \
-rxPower 1.0 \
-idlePower 1.0 \
-sleepPower 0.001 \
          -channel       $chan \
          -agentTrace    ON \
          -routerTrace   ON \
          -macTrace      ON \
          -movementTrace ON
```

Appendix B: Running simulation and computing result using awk script



```
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ ns eesmac.tcl
num_nodes is set 30
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
INITIALIZE THE LIST xListHead
Starting Simulation...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
node: 22 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 5 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 22 ..............data sent Uni............
node: 15 ..............data sent Uni............
node: 7 ..............data sent Uni............
node: 14 ..............data sent Uni............
node: 14 ..............data sent Uni............
```



```
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac20.tr
  610.308463
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac30.tr
  1990.721127
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac40.tr
  3016.564175
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac50.tr
  2978.361260
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac60.tr
  3997.122064
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac70.tr
  4463.034319
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac80.tr
  4761.903821
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac90.tr
  4811.432326
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f Throughput.awk eesmac100.tr
   4991.123099
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  20.118982
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  17.120082
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  18.361282
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  19.649182
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  21.084082
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  22.489082
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  17.009082
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  18.501182
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ gawk -f average_delay.awk eesmac.tr
 Average end to end delay  16.814082
vboxuser@Ubuntu:~/Downloads/TD/EESMAC$ _
```