ENERGY UTILIZATION AND SCHEDULING OF CONVERGECAST IN MULTICHANNEL WIRELESS SENSOR NETWORK

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Electronics and Communication Engineering

By

Vishav Kapoor

Registration Number: 41800923

Supervised By

Dr. Daljeet Singh (23804)

Assistant Professor, School of Electronics and Electrical Engineering

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2024

DECLARATION

I, hereby declared that the presented work in the thesis entitled "Energy Utilization and Scheduling of Convergecast in Multichannel Wireless Sensor Network" in fulfilment of degree of **Doctor of Philosophy (Ph.D.)** is outcome of research work carried out by me under the supervision of Dr. Daljeet Singh, working as Assistant Professor, in the School of Electronics and Electrical Engineering of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Vishav Kapoor

Registration No.: 41800923

Department/school: Electronics and Communication

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled “Energy Utilization and Scheduling of Convergecast in Multichannel Wireless Sensor Network” submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the School of Electronics and Electrical Engineering, is a research work carried out by Vishav Kapoor, Registration No. 41800923, is bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

Daljeet
23804

(Signature of Supervisor)

Name of supervisor: Dr. Daljeet Singh

Designation: Assistant Professor

Department/School: School of Electronics and Electrical Engineering

University: Lovely Professional University

Abstract

Convergecast is a key data gathering process in multichannel wireless sensor network (WSN). Its goal is to receive and send data from multiple sensor nodes to a central sink node or base station. The need for convergecast comes from the fact that each sensor node has limited energy, memory, and bandwidth. This means that data needs to be combined to reduce transmission overhead and energy use. But convergecast in multichannel WSN is associated with problems like channel interference, scaling, different types of nodes, and security issues. Multichannel WSN solves some of these problems by sending data over more than one channel, which can cut down on congestion and increase the network's capacity. But convergecast multichannel WSN's channel distribution brings up problems like channel interference, route overhead, and network dynamics. Also, the energy savings of convergecast multichannel WSN is significant if the network to last as long as possible.

To overcome the issues mentioned above, several approaches can be adopted. For problems with channel interference and scalability, distributed algorithms can be used for the allocation of channels and aggregation of data for reducing the amount of central control and make the system more flexible. Adaptive data compression and routing techniques that take into account the capabilities of each node can be used to deal with the various nodes. Optimizing energy use can be done with sleep/wake scheduling algorithms, low-power communication protocols, and energy harvesting techniques.

In this dissertation, a convergecast approach for multichannel WSN which is energy efficient has been presented. It includes the study and analysis of various convergecast techniques, channel allocation methods, and data aggregation procedures based on the network parameters. The other contribution is the introduction of fuzzy- based sleep scheduling mechanism that would activate or sleep sensors as needed to decrease energy usage. Performance indicators like throughput, packet loss, lifespan, energy consumption, and latency are being used to assess the effectiveness of the proposed approach with the existing approach. Another contribution is the introduction of a belief score-based approach for effective slot allocation in a multi-channel environment. By limiting participation in the convergecast process to trusted nodes, a belief score authentication scheme improves the security of data aggregation. Throughput, packet loss, lifespan, energy consumption, and latency are performance indicators that are being used to evaluate the performance of the suggested approach. The development of an architecture for the lightweight multichannel MAC protocol for convergecast WSNs while gathering data is the thesis's fourth contribution. It shows that there is an effective method for convergecast WSN channel allocation in a multichannel environment.

Acknowledgments

I am indebted to my Ph.D. guide Dr. Daljeet Singh for his gracious encouragement and very valuable constructive criticism that has driven me to carry out the research work for this dissertation successfully. I am deeply grateful to the Ph.D. committee for their support and encouragement in carrying out my Ph.D. work.

I am extremely thankful to the management of Lovely Professional University for providing me with infrastructural facilities to work, without which this work would not have been possible.

Finally, I express my deep sense of gratitude to my parents, my wife, my daughter, and almighty God for their blessings without which this thesis was not possible.

Vishav Kapoor

List of Publications

Journal

- Vishav Kapoor and Daljeet Singh, “FBESSM: Fuzzy Based Energy Efficient Sleep Scheduling Mechanism for Convergecast in Wireless Sensor Networks,” International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 9S, 2023.
- Vishav Kapoor and Daljeet Singh, “EMCSABA: Efficient Multi Channel and Slot Allocation with Belief Score Authentication for Wireless Sensor Networks,” submitted to IEEE Sensors Letters (Communicated).

International Conferences

- Vishav Kapoor and Daljeet Singh, “Scheduling Techniques for Convergecast in Multichannel Wireless Sensor Networks: A Review and Issues,” in 6th International Conference on Computing Sciences 2022, Lovely Professional University, Punjab, Nov. 2022,

- Vishav Kapoor and Daljeet Singh, “A Comparative Review on Channel Allocation and Data Aggregation Techniques for Convergecast in Multichannel Wireless Sensor Networks,” in 3rd International Conference on Information Technology 2023 (InCITe 2023), Amity University, Mar. 2023.

TABLE OF CONTENTS

Title	**Page No.**
Abstract	iii- iv
List of Publications	vi
Table of Contents	vii-x
List of Tables	xi
List of Figures	xii-xiii
List of Abbreviation	xiv-xv
CHAPTER 1: Introduction	1-19
1.1 Introduction to WSN	1
1.2 Evolution of WSN	2
1.3 Applications of WSN	4
1.4 Types of WSN	6
1.4.1 Based on Node Similarity	6
1.4.2 Based on Node Position	7
1.4.3 Based on Medium of Operation	7
1.4.4 Based on Data Flow	8
1.4.5 Based on Number of Channels	8
1.5 System Model of WSN	9
1.6 Convergecast in WSN	11
1.7 Types of Convergecast WSNs	12
1.8 Convergecast Multichannel WSN	13
1.9 Challenges in Convergecast Multichannel WSN	13
1.9.1 Issues Related to Energy Utilization	14

1.9.2 Issues Related to Channel Allocation 15
1.9.3 Issues Related to Data Aggregation 15
1.9.4 Issues Related to Scheduling Techniques 16
1.10 Performance Evaluation of WSN 17
1.11 Thesis Organization 18
CHAPTER 2: Literature Work 20-40
2.1 Motivation of Work 20
2.2 Survey of Convergecast WSNs 21
2.3 Survey of Multichannel Convergecast WSNs 23
2.3.1 Multichannel MAC 23
2.3.2 Survey of Multi-Channel Routing 26
2.4 Survey of Energy Utilization in WSNs 29
2.5 Survey of Data Aggregation 33
2.6 Survey of Scheduling Techniques 34
2.7 Comparative Evaluation Based on Literature Work 36
2.8 Objectives of the Proposed Work 39
2.9 Research Methodology 39
CHAPTER 3: FBESSM: A Fuzzy Based Energy Efficient Sleep Scheduling Mechanism 41-70
3.1 Framework of Convergecast in Multichannel Wireless Sensor Network 41
3.1.1 IEEE 802. 15.4 Protocol 43
3.2 Creation of OSCAR Algorithm 48
3.2.1 Limitations of OSCAR 51

3.3 Proposed FBESSM Approach 52

3.4 Performance Analysis 57

3.4.1 Simulation Results 57

3.4.2 Energy Model 60

3.4.3 Metrics for Evaluation 60

3.4.4 Comparative Evaluation 61

3.5 Summary 68

CHAPTER 4: EMCSABA: Efficient Multi Channel and Slot Allocation with Belief Score Authentication 70-90

4.1 Slot Allocation Process in Multichannel WSN 70

4.2 Proposed Scheme for Efficient Slot Allocation 71

4.2.1 Efficient Multichannel and Slot Allocation 72

4.2.2 Efficient Slot Acquisition 75

4.3 Performance analysis of EMCSABA algorithm 78

4.3.1 Comparative analysis based on node count 78

4.3.2 Comparative analysis based on simulation time 80

4.3.3 Comparative Evaluation with Existing Techniques 86

4.4 Summary 90

CHAPTER 5: MLMP: Multichannel Lightweight MAC Protocol 91-105

5.1 Multichannel in Convergecast WSN 91

5.1.1 Need for Multichannel in Convergecast WSN 91

5.1.2 Challenges in Multichannel Convergecast WSNs 92

5.2 Proposed Lightweight Multichannel MAC Scheme 94

5.2.1 Proposed Algorithm for MLMP 95

5.3 Comparative Analysis 99

5.4 Summary 104

CHAPTER 6: Conclusion and Future Work 106-108

6.1 Conclusion 106

6.2 Future Works 107

References 109-119

LIST OF TABLES

Table No	Title	Page No
2.1	Comparative evaluation of channel allocation techniques on WSN	27
2.2	Comparison of various approaches in Convergecast WSNs	30
2.3	Comparative evaluation of data aggregation techniques on WSN	37
2.4	Metric based evaluation of Convergecast techniques	38
3.1	Simulation parameters for the proposed FBESSM approach	59
3.2	Metric based evaluation for the OSCAR technique	61
3.3	Metric based evaluation for the proposed FBESSM technique	62
4.1	Comparative evaluation of EMCSABA with existing techniques based on PDR	87
4.2	Comparative evaluation of EMCSABA with existing techniques based on EED	88
5.1	EC variation with node count for MLMP	99
5.2	PDR variation with node count for MLMP	102
5.3	RE variation with node count for MLMP	102

LIST OF FIGURES

Figure No	Title	Page No
1.1	WSN Classification	7
1.2	System Model of a Sensor Node	9
1.3	Architecture of WSN	11
1.4	Convergecast WSN Architecture	12
3.1	Details of a log created for packet transmission/ reception time stamps, route establishment, and routing table updates.	45
3.2	IEEE 802.15.4 protocol	46
3.3	Multichannel selection for Convergecast	47
3.4	Multichannel WSN nam file output	47
3.5	Flowchart of OSCAR approach	49
3.6	Working of FBESSM technique	58
3.7	Variation of EED under the node count	63
3.8	Variation of EC under the node count	63
3.9	Variation of NLT under the node count	64
3.10	Variation of PDR under the node count	65
3.11	Variation of packet loss under the node count	66
3.12	Variation of throughput under the node count	66
4.1	Variation of EC under the node count	80
4.2	Variation of NLT under the node count	81
4.3	Variation of PDR under the node count	81

4.4	Variation of RE under the node count	82
4.5	Variation of delay under the simulation time	83
4.6	Variation of EC under the simulation time	83
4.7	Variation of packet loss under the simulation time	85
4.8	Variation of PDR under the simulation time	86
4.9	Variation of throughput under the simulation time	87
5.1	Proposed working of lightweight multichannel MAC protocol	98
5.2	EC based comparative analysis by varying the node count	101
5.3	PDR based comparative analysis by varying the node count	101
5.4	RE based comparative analysis by varying the node count	103

LIST OF ABBREVIATIONS

MEMS	Micro Electro Mechanical Systems
LEACH	Low-Energy Adaptive Clustering Hierarchy
MSNHR	Mobile Sensor Networks Routing
WSN	Wireless Sensor Network
STC	Spanning Tree Convergecast
MWSN	Multichannel Wireless Sensor Network
FCA	Fixed Channel Allocation
DCA	Dynamic Channel Allocation
TDMA	Time Division Multiple Access
FDMA	Frequency Division Multiple Access
CDMA	Code Division Multiple Access
QoS	Quality of Service
ESD	Electronically Switched Directional
MAC	Media Access Control
TSCH	Time Slotted Channel Hopping
PCSS	Predictive Channel Scanning and Switching
HMC	Hybrid Multi Channel
MCC	Multi Channel Collection
CODA	Collision Avoidance and Detection
CH	Cluster Head
HEED	Hybrid Energy Efficient Distributed Clustering
MOFCA	Multi Objective Fuzzy Clustering Algorithm
MODESA	Multichannel Optimized Delay Time Slot Assignment
CFAS	Collision Free Advertising Scheduling

FBESSM	Fuzzy Based Sleep Scheduling Mechanism
ACK	Acknowledgement
CCA	Clear Channel Assessment
CW	Contention Window
RPL	Low Power and Lossy Networks
PDR	Packet Delivery Ratio
NLT	Network Life Time
EED	End End Delay
RREQ	Route Request
DSR	Dynamic Source Routing
RREP	Route Reply

CHAPTER 1

INTRODUCTION

1.1 Introduction to WSN

The development of wireless communication and Micro Electro Mechanical Systems (MEMS) has made it possible to develop sensor nodes that are affordable, small, less power-consuming, and can perform multiple tasks. They can also communicate seamlessly through a wireless medium. The research fraternity is interested in these qualities of Wireless Sensor Networks (WSNs) make them useful in various applications [1]. A WSN is made up of one or more sink nodes and geographically isolated sensors. Sensors produce sensual data and constantly monitor factors including temperature, pulsation, and motion.

WSNs ability to monitor and gather data from the physical world has made them of utmost importance in contemporary society and revolutionized several fields. Numerous tiny, interconnected sensors make up these networks, working together to wirelessly collect and transmit data. WSNs have demonstrated their value in a variety of industries, including healthcare, industrial automation, and smart cities. WSNs are essential for tracking variables like temperature, humidity, air quality, and water levels in environmental monitoring [2]. For instance, by providing real-time information on soil moisture and weather conditions, these networks support agriculture by optimizing irrigation and crop management. WSNs aid in early warning systems for earthquakes, tsunamis, and forest fires in disaster management, enabling quick response and reducing casualties.

The research work aims to improve the performance of convergecast multichannel WSNs by utilizing novel techniques and architectural proposals. The fundamental motivation for this research is to address the substantial difficulties that arise in the context of WSN energy efficiency, reliability, and communication quality. To achieve this task, firstly, an energy utilization algorithm is developed for refining the efficiency of the WSN system. This technique improves the overall energy proficiency of the system by optimizing energy consumption throughout the convergecast process. Further, the proposed technique also extends the network lifetime and data-gathering of every sensor node [3].

Furthermore, a scheduling technique is developed to reduce collisions of data packets, which is another critical issue influencing the reliability of WSNs. This technique reduces packet collisions by deliberately arranging time slots and communication channels to suit a given scenario. Thereafter, the work is extended to develop multichannel communication architecture in response to the increasing demand for robust communication. This architecture uses multiple communication methods improving the overall performance of the network which may result in providing good communication networks and effective data transfer in the presence of numerous types of interference [4].

The importance of this research stems from its potential to change the way convergecast multichannel WSNs operate. This study contributes to more sustainable and dependable WSNs by proposing energy-efficient algorithms, collision avoidance measures, and enhanced communication topologies for convergecast multichannel WSNs. The presented methodologies provide realistic answers to real-world problems, with applications in various areas such as environmental monitoring, health care and industry automation, where reliable and energy-efficient data collection is critical.

1.2 Evolution of WSN

WSNs have undergone a remarkable evolution in the last few decades in terms of technological advancement and application scenarios. Initial developments in WSNs were focused towards military and academic research, but they are now widely used across a variety of fields. With the advancements in technology, miniaturization has resulted in more compact, energy-efficient sensors that can gather data autonomously. Zigbee and Bluetooth have further revolutionized WSNs by enabling seamless data transmission and extending network lifespan due to their low energy consumption and improved performance. WSNs are useful in healthcare for patient monitoring, in agriculture for precision farming, and in industrial automation for process optimization [5]. WSNs can provide useful insights from the gathered data using the efficient integration of machine learning and data analytics which improves the system's capacity for making decisions. 5G technology, power computing, and cloud computing have enabled high-speed, low-latency communication in WSNs. The main developments in WSN with their technical specifications are as follows:

Early Research (1960s–1980s): The groundwork for wireless communication and data transmission was laid by early experiments done by the United States army project named Sound Surveillance System (SOSUS) [4].

Development of Sensor Nodes (1990s): In this era, the sensor nodes gained processing abilities and constrained energy resources, enabling wireless communication and simple data processing. One such example is the University of California, Los Angeles (UCLA) Wireless Integrated Network Sensors [5].

Ad Hoc Networking (Early 2000s): The use of ad hoc networking produced self-organizing WSNs, which allowed nodes to cooperate and establish communication without centralized control.

Low-Power Protocols (Early 2000s): The development of protocols like Zigbee and Bluetooth [6] made it possible to transfer data efficiently while maintaining node operation for extended periods.

Applications in Surveillance and Monitoring (Early 2000s): WSNs were used for environmental monitoring, habitat tracking, and military surveillance, proving their utility outside of the academic world.

Wearable Sensors in Healthcare (Early 2010s): Wearable sensors for patient monitoring were the first application of WSNs in healthcare, improving remote healthcare management and diagnostics.

Energy Harvesting (Mid-2010s): Energy-efficient methods, such as solar and kinetic energy harvesting [7], increased the duration of sensor nodes and decreased the frequency of battery replacements.

Efficient routing protocols (Late 2010s): Cutting-edge routing algorithms optimized data transmission paths, saving energy and extending network lifespan [8].

Integration with IoT and Cloud (Present): WSNs and Internet of Things (IoT) are combined to transform industries like smart cities and industrial automation by enabling cloud integration, data analytics, and seamless interaction with other interconnected devices.

1.3 Applications of WSN

WSN can be used for numerous applications such as target tracking, health monitoring, control vehicles, and keeping an eye on the military. The statistics gathered by a WSN is sent to an operator who is responsible of handling the data and processing it so that smart decisions can be made. Traditionally, these sensors were linked by a guided medium that resulted in a complicated network of wires that was difficult to maintain and troubleshoot and resulted in an unreliable system. WSNs have a wide range of applications in a variety of fields, which gives them tremendous significance. WSNs are used in various industries, including agriculture, healthcare, industrial automation, and environmental monitoring [9].

Environmental Monitoring: By providing real-time data on variables like temperature, humidity, air quality, and pollution levels, WSNs have revolutionized environmental monitoring. Scientists and policymakers can monitor climate change, precisely track natural disasters, and ensure proper management of ecosystems through the optimal use of WSN. Zigbee and IEEE 802.15.4 are two common standard protocols used by WSNs for environmental monitoring [9] which are available everywhere and are compatible with most of the devices working in the 2.4 GHz Industrial Scientific and Medical (ISM) band [10].

Healthcare: WSNs have many advantages for the healthcare industry as well because wearable sensors can track vital signs and assist doctors in remotely monitoring the health of patients [11]. This is especially important for elderly people and those who suffer from chronic illness. WSNs have enabled telemedicine, wearable health devices, and remote patient monitoring in the healthcare industry. Wearable technology has sensors that can track vital signs and allow doctors to remotely check on their patients. These networks follow Continua Health Alliance regulations and IEEE 11073 medical standards [12]. Medical device communications are restricted to specific frequency bands, such as the Medical Body Area Network (MBAN) spectrum at 2.4 GHz [13].

Industrial Automation: By enabling condition monitoring, proactive maintenance, and process improvement, WSNs are essential to industrial automation. Industrial standards like WirelessHART and ISA100.11a [14], which are designed for dependability and performance in industrial settings, are used in these networks. Based on the particular needs of the industrial application, frequency bands like the 900 MHz and 2.4 GHz ISM bands are used [15]. WSNs improve efficiency and safety in industries by enabling predictive maintenance of machinery through ongoing equipment performance monitoring and anomaly detection. Power plants and aircraft automation, where thousands of sensors monitor temperature, position, and pressure in confined spaces, represent other significant use cases for WSNs. WSNs are essential to the implementation of smart cities because they make it possible to manage waste, traffic, and energy consumption more effectively. Numerous applications for smart cities can support the communication requirements of standards like IEEE 802.15.4 and LoRaWAN. ISM bands with frequencies like 868 MHz, 915 MHz, and sub-GHz offer a wide range of coverage and penetration in urban settings [16].

Agriculture: WSNs help betterment in agriculture by providing actual information on crop health, soil moistness, and temperature. The use of these networks helps to improve crop management, fertilization, and irrigation techniques. In agricultural WSNs, standards like IEEE 802.15.4 and LoRaWAN are frequently used for agricultural applications [17]. For long-distance communication in rural areas, frequency bands like the 868 MHz and

915 MHz ISM bands are used [16].

Wildlife Monitoring: By observing animal behaviour, migratory patterns, and habitat conditions, WSNs help in wildlife monitoring. These networks support environmentalists in their efforts to comprehend and safeguard threatened species. Wildlife monitoring WSNs use both proprietary protocols and standards like IEEE 802.15.4. These standards can be used with frequency bands like the 2.4 GHz ISM band [18].

Home automation: WSN makes it possible to connect smart devices for lighting, security, energy management, and other purposes. Zigbee and Z-Wave standards guarantee seamless communication between products made by various manufacturers [19]. The ISM bands are used for home automation applications.

1.4 Types of WSN

WSNs come in a variety of shapes and designs that are reformed for various applications. These variations in WSN types are created to maximize performance, energy efficiency, coverage, and data delivery. WSNs can be classified based on a variety of features and a comprehensive classification of WSNs is presented below.

1.4.1 Based on Node Similarity

Homogeneous WSN: Homogeneous WSNs consist of multiple sensor nodes with identical hardware and functionalities. This simplicity makes deployment, data gathering, and maintenance simpler. Uniform data sensing is necessary, such as when monitoring temperature, humidity, and light intensity indoors, homogeneous WSNs are used [3].

Heterogeneous WSNs: Heterogeneous WSNs comprise sensor nodes with a range of functions, processing speeds, and sensor types. Due to the ability of nodes with specialized sensors to collect particular data, this kind of network enables the collection of data that is both more complex and diverse. Applications like disaster management, where various sensors can monitor various aspects of a disaster-affected area, frequently use heterogeneous WSNs.

1.4.2 Based on Node Position

Mobile WSNs: They can be used for applications requiring dynamic data collection because they have nodes that can move around in a given area. These networks are used in situations like monitoring rapidly changing environments, guiding autonomous vehicles, and tracking the movement of wildlife.

Static WSNs: It refers to a WSN with fixed sensor nodes. The deployment and maintenance of stationary WSNs are both simple. Positioning the various sensor nodes in the most efficient location allows for the placement of more nodes in a smaller area.This WSN type's topology can be modified automatically. Network partition happens if any node fails [6]. Fig. 1.1 shows WSN types and their further classification.

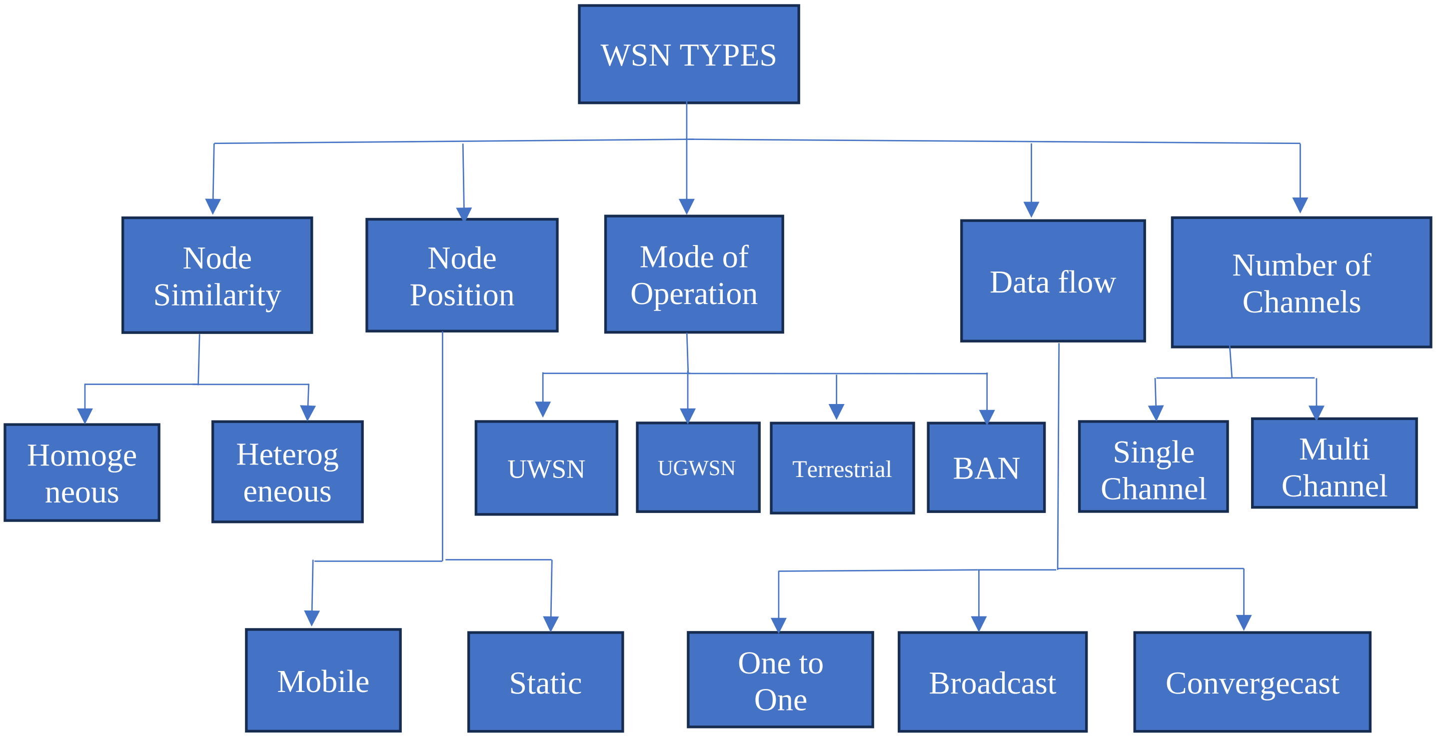


Fig 1.1: WSN classification

1.4.3 Based on Medium of Operation

Underwater WSNs: Data gathering in aquatic environments is made possible by underwater wireless sensor networks, or UWSNs. As complications are associated with underwater signal propagation, these networks employ specialized acoustic or optical communication methods. Applications include underwater pipeline monitoring and oceanographic research.

Underground WSN (UGWSN): UGWSNs are set up below the surface of the earth for uses like mining, tunnel monitoring, and subsurface environmental assessment. Due to the limited signal propagation in rock and soil, these networks face communication difficulties [8].

Terrestrial WSN: Terrestrial WSNs are used for industrial automation, precision farming, and environmental monitoring and operate on the ground. Because of their flexibility, these networks may use a range of communication technologies that depend on the needs of the application.

Body Area Networks (BANs): BANs use wearable sensors to track vital signs, medical conditions, and physical activity on or inside the human body. This is necessary for medical solicitations like telemedicine and healthcare examinations because they allow for real time health tracking [5].

1.4.4 Based on Data Flow

One to One: This is the conventional way to send data from one node to another. It is very rarely used now for its inefficiencies like bandwidth and timing constraints.

Broadcast: Spreading a message that was originated by a source node to each node that can receive it in the network is known as broadcasting. Networks that propagate the information in this way are broadcast.

Convergecast: The process of disseminating data between various sensor nodes is referred to as broadcast, and the process of collecting information from sensor nodes is referred to as convergecast. In the current study, we are utilizing the convergecast feature of WSN [2].

1.4.5 Based on Number of Channels

Single-Channel: It describes a particular class of WSN in which every sensor node utilizes the same radio frequency channel for communication.

Multi-Channel: It is a type of WSN in which sensor nodes communicate across numerous radio frequency channels rather than just one. This method is intended to reduce network interference and congestion while improving overall network performance. In the current study, we are utilizing the multichannel nature of WSN.

1.5 System Model of WSN

A sensor node consists of four main parts i.e.: a sensing unit that collects data, a processing unit that processes data, a communication unit that acts as a transmitter and receiver for data communication within the network, and a main power unit consisting of two subparts, regenerative power unit and backup power. The regenerative unit and the backup power unit usually have less power storage capacity due to size, weight, and cost restrictions [20]. Fig. 1.2 shows the system model of a wireless sensor node.

The node is connected to the network by a communication unit. Any wireless sensing node must minimize system power consumption as a critical component. The node programs an algorithm to determine when to send data based on the perceived event, thereby reducing the power consumption of the sensor.

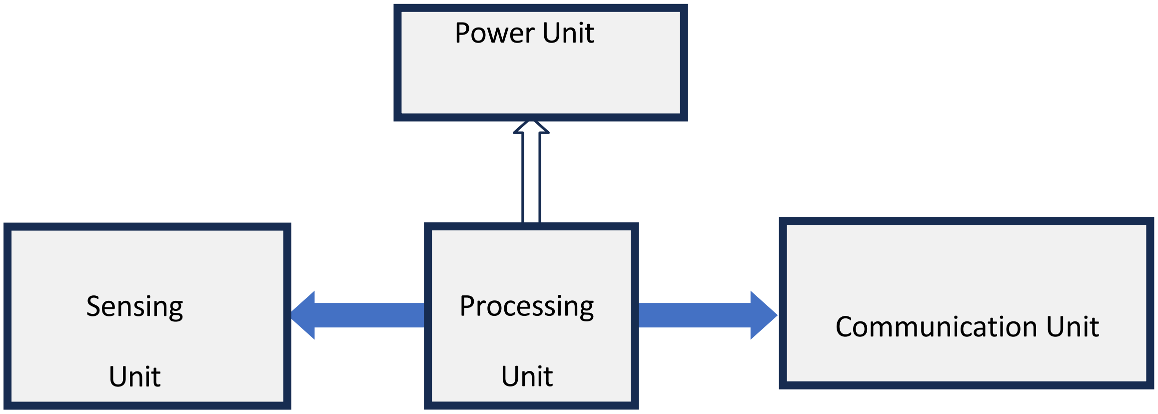


Fig 1.2: System model of a sensor node

Most sensor networks have already switched to wireless links as they can be moved around and are easier to set up. On the other hand, the major problem with WSN is limited power supply through onboard battery, which makes it an energy-constrained network and limits its capabilities in terms of life span and transmission power [22, 23].

Therefore, most of the WSNs are short-range networks with a small amount of bandwidth. These WSNs make use of multi-hop architecture for communication. Fig.1.2 shows the architecture of a typical WSN [24]. Additionally, these sensor nodes include several vital parts that are necessary for their operation, including a sensor that can detect an event and convert sensed data into electrical signals. The data will then be sent to the microcontroller for processing into an electrical signal, which the transceiver will then receive to establish communication.

Sensor Nodes: The supervised area is covered by various sensor nodes. Each node has sensors to gather information. Nodes have limited resources, including energy, memory, and processing power. They can wirelessly communicate with the sink and other nearby nodes. Data can be sensed, processed, and transmitted by nodes. Nodes can be either stationary or mobile, depending on the application. There are two types of nodes in a WSN: gateways and remote nodes. WSN usually uses a "many-to-one" method of communication, since many sensor nodes are set up in one place [26-27]. Each sensor node gathers information about its surroundings and sends that information to a gateway [25].

Communication Channel: Channels for wireless communication make it easier to transfer data between nodes and the sink. Because of the wireless medium, channels may encounter fading, noise, and interference with a unique range, data rate, and power consumption characteristics (such as Zigbee, Bluetooth, and Wi-Fi). The consistency of data transmission is impacted by the quality of the communication channels.

Base/Sink Station: The sink, also referred to as the base station or gateway, is the central node responsible for data collection and coordination. Between the WSN and outward systems or networks, it works as a gateway or path. It gathers and aggregates data from sensor nodes, processes it as needed, and then transmits it to higher-level systems. To improve data collection, the sink may employ sophisticated routing algorithms [22].

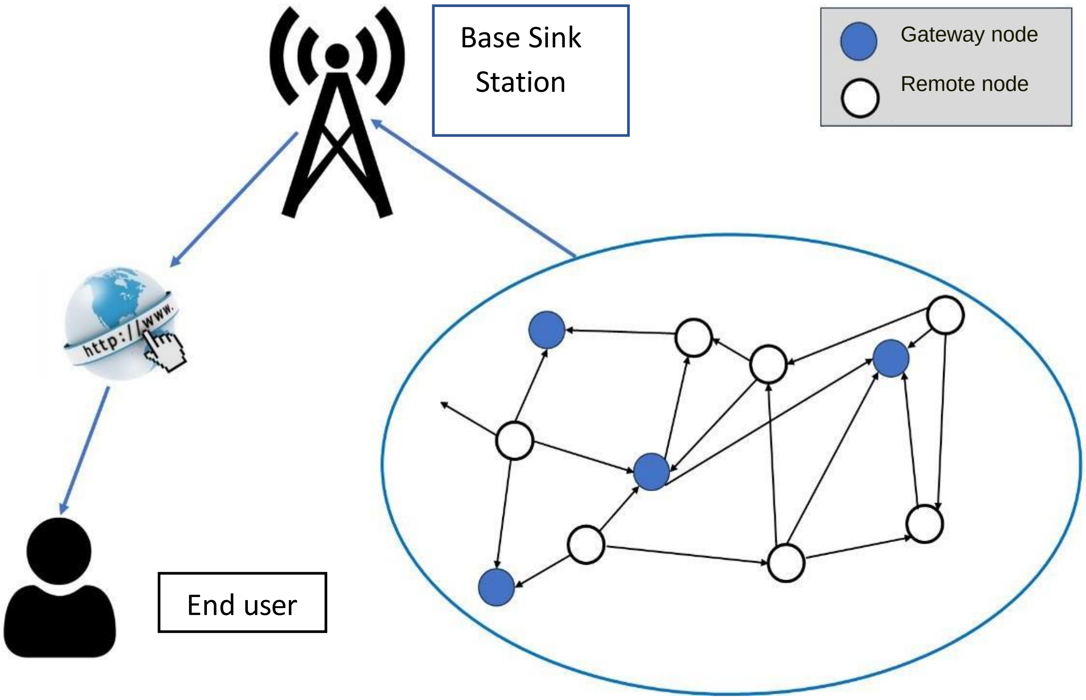


Fig. 1.3: Architecture of WSN The components of a WSN

1.6 Convergecast in WSN

Convergecast is a communication pattern used in WSNs to transmit data from multiple nodes to a single node, typically a base station. The word "Convergecast" refers that data is converging from more than one node to a single node. Convergecast is the primary method of communication between nodes of a WSN [28] and their data to the sink after sensing some data or receiving a query. Nodes can transmit their data unaltered, or they can aggregate it at each sensor node that passes through a sink node.

This process of getting data from one place to another takes time and energy, so a data transmission scheduling approach is needed to handle the data traffic and cut down on the amount of energy used by each node to increase network lifetime. Another important parameter in data converge casting is to reduce delays and ensure the successful delivery of data. Fig. 1.4 shows the architecture of a typical Convergecast WSN. It can be visualized from Fig. 1.4 that when signals are moved from source to sink node, the information from multiple nodes is taken by a single node in between and then disseminated to the next source node [26].

1.7 Types of Convergecast WSNs

In multichannel wireless sensor networks (WSNs), several types of convergecast techniques can be used for data gathering from sensor nodes [29]. Here are three common types of convergecast in multichannel WSN:

Single path convergecast: In a single path convergecast, the data is collected from sensor nodes using a single way for transmission from the source to the sink node. The pathway is selected depend on the available channels and the quality of the link between various nodes. This method is simple and easy to set up, but it can slow down and clog up a network, especially if it is large or full of devices.

Multi path convergecast: In a multi-path convergecast, data or signals are transmitted from source nodes to sink nodes using more than one path. This approach can improve network performance and reliability by reducing congestion and providing redundancies. However, it requires more complex routing algorithms and can result in increased energy consumption due to the need for multiple transmissions.

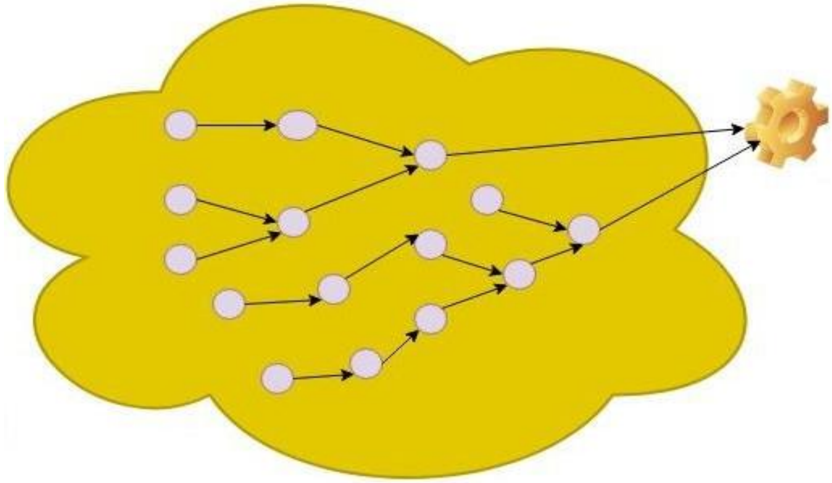


Fig. 1.4: Convergecast WSN Architecture

Hierarchical convergecast: In a hierarchical convergecast, the sensor nodes are organized into various multiple levels. Each level oversees collecting information from an internal set of nodes and forwarding it to the next level. The sink node is typically located at the highest level and collects data from all lower levels. This approach can improve network scalability and reduce congestion, but it requires more complex network management and communication protocols. Overall, the choice of convergecast technique depends on what the application and environment need and how they work. When choosing a convergecast technique for a multichannel WSN, one should think about things like the network’s size, density, and energy limits, as well as how reliable the data needs to be and how long it needs to take to get there [24].

1.8 Convergecast Multichannel WSN

Convergecast WSNs are networks that are designed to collect data from a great count of sensor nodes and transmit it to a central sink node for processing. In a convergecast WSN, the sink node initiates communication with the sensor nodes. Due to many sensor nodes, convergecast WSNs are prone to various forms, which leads to generation of congestion and contention, which can lead to reduced network performance. Multichannel MAC protocols can be used in convergecast WSNs to address these issues and improve network efficiency [6]. In a multichannel convergecast WSN, every sensor node is allocated a precise channel for communication with the sink node. The sink node uses a different channel for communication with each sensor node, enabling concurrent transmissions and reducing contention on a single channel. Multichannel protocols also allow for better utilization of available bandwidth, as nodes can switch to a less congested channel if necessary.

For example, consider a convergecast WSN that is used to monitor a large industrial plant. The sensor nodes are arranged throughout the plant and collect data on different parameters, such as temperature, humidity, and pressure. The data is transmitted to a central sink node, which processes the data and sends control signals back to the plant. In a single-channel protocol, the sensor nodes would compete for the same channel, leading to contention and reduced network performance. By using a multichannel protocol, every sensor node can be assigned to a definite channel, reducing collisions and improving the overall productivity of the system [8].

Another example of a multichannel convergecast WSN is a smart city network. In a smart city network, sensors are deployed in the city to check different parameters. The data is transmitted to a central sink node, which processes the data and sends control signals back to the city infrastructure. In a single-channel protocol, the large number of sensor nodes could lead to congestion and interference on the channel. By using a multichannel protocol, the sensor nodes can be assigned to different channels, enabling concurrent transmissions and reducing contention on a single channel [23].

1.9 Challenges in Convergecast Multichannel WSN

Multichannel WSNs are complex systems that can face several issues and challenges which are as follows:

1.9.1. Issues Related to Energy Utilization

Energy efficiency is vital parameter in convergecast MWSNs. Energy-efficient convergecast is necessary to persist the lifetime of the network, minimize energy consumption, and ensure reliable data transmission [18]. However, there are several challenges and issues associated with energy efficiency in convergecast multichannel WSNs. Some of these challenges include:

Interference between channels: In WSNs with more than one channel, interference between channels can cause collisions and packet loss, which use more energy. Energy-efficient convergecast algorithms must affects the channel interference.

Node mobility: The mobility of sensor nodes can cause changes in channel conditions and topology, leading to increased energy consumption. Energy-efficient convergecast algorithms must adapt to changes in channel conditions and topology to ensure reliable data transmission.

Routing overhead: Routing protocols in convergent multichannel WSNs can add a lot of extra work, which makes them use more energy. Energy-efficient convergecast algorithms must minimize routing overhead to conserve energy.

Network congestion: When there is too much traffic on a network, it can use more energy and last less long. Energy-efficient convergecast algorithms need to deal with network congestion to make sure data is sent reliably and to use as little energy as possible.

Security: Security is a critical concern in convergecast multichannel WSNs. Energy-efficient convergecast algorithms must be designed to ensure secure data transmission while minimizing energy consumption.

1.9.2 Issues Related to Channel Allocation

While channel allocation can improve the performance and efficiency of convergecast Multichannel WSNs, several issues and challenges need to be addressed, including:

Channel Interference: Channel interference is one of the biggest problems with channel allocation. This problem can get worse in networks with a lot of people or a lot of moving parts, where interference patterns can change quickly and in unpredictable ways [19].

Channel Assignment Overhead: Another problem is the overhead that comes with channel assignment, which can use up a lot of network resources, especially in centralized algorithms. This can include channel scanning, synchronization, and signaling, as well as the computation and communication needed for channel allocation.

Scalability: Another problem is that some channel allocation algorithms may not be good for large networks or networks with a lot of changes. For instance, centralized algorithms might need a lot of coordination and communication, which might not be possible in large networks where people move around a lot.

Adaptability: Channel allocation algorithms should be able to handle node failures or moving nodes and this needs algorithms that can keep an eye on the network and change how channels are used on the fly. They also need to be able to reassign channels when nodes stop working or leave the network [23].

1.9.3 Issues Related to Data Aggregation

Data accumulation is an important technique in convergecast MWSN, but there are several issues and challenges associated with it. Some of the major issues are discussed below:

Data accuracy: The accuracy of data is a critical issue in data aggregation. As data is aggregated from multiple sources, there is a chance of errors in data transmission or collection. These errors can cause data inaccuracies, which can expressively affect the performance of the WSN.

Data security: Data security is another significant issue in data aggregation. Since data is transmitted over wireless channels, it is vulnerable to interception, tampering, and eavesdropping. The aggregation process also involves the collection of sensitive data from multiple nodes, which can be a potential target for attackers. Therefore, proper encryption techniques must be used to secure the data [17].

Scalability: If the count of nodes in WSN increases the data aggregation process becomes challenging. Therefore, scalable data aggregation techniques are required to handle large-scale WSNs.

Heterogeneity of nodes: In WSNs, nodes can have different features and abilities, such as different battery sizes, processing speeds, and communication ranges. These differences can create challenges in data aggregation, as some nodes may be overloaded with data, while others may not have enough data to transmit.

Routing: To make sure that data enters the sink node in the best way possible, data aggregation needs efficient routing mechanisms. Traditional routing protocols might not be good for data aggregation because they might not take into account the process of aggregation [14].

1.9.4 Issues Related to Scheduling Techniques

Scheduling techniques can improve the efficiency of communication and lower the energy consumption in convergecast Multichannel WSN, they also present some issues and challenges. Some of these include:

Overhead: Some scheduling techniques can cause additional network overhead. For example, reservation-based MAC protocols require nodes to send reservation messages to reserve a time slot or frequency band, which increases communication overhead and reduces the available bandwidth for data transmission.

Scalability: Some scheduling methods might not work well in big networks with a lot of nodes. For example, TDMA requires coordination between nodes to assign time slots, and the coordination overhead goes up as the number of nodes goes up [12].

Dynamic environments: Scheduling techniques may not be effective in dynamic environments. For example, reservation-based MAC protocols may not be suitable in such environments, as they require fixed time slots or frequency bands.

Complexity: Some scheduling methods may be hard to set up and manage, especially in large networks. This complexity can result in increased implementation and maintenance costs.

1.10 Performance Evaluation of WSN

The following parameters have been proposed in the literature and utilized in this work for evaluating the performance of convergecast and multichannel WSNs:

Latency: The time between the first input and the first output is called latency. Milliseconds are often used to measure this amount of time. Transport delay is another name for latency. Few researchers discriminate between latency and transport delay in terms of the additional time delay of a system over and above the reaction time.

Energy Consumption: This is the crucial factor for WSN as sensor nodes are energy limited and have non-replaceable batteries. This parameter depends upon various aspects such as routing of data, active and idle time of sensor nodes and distance from the sink node.

Throughput: Throughput is a parameter in which data can be transferred from one position to another within a specified time. It is generally used to determine the performance of network connections.

Packet Loss Ratio: The packet loss ratio symbolizes the count of lost packets compared to the total data packets sent. Each packet has a deadline before which it must be executed, and if this is not possible, the data packet may get lost.

1.11 Thesis Organization

The first chapter is oriented to introduce the reader to the convergecast in multichannel WSN. A thorough introduction is presented on convergecast WSNs, Evolution in WSN, Applications of WSN, Types of WSN, System Model of WSN, Convergecast in WSN, and its types and challenges along with multichannel WSN. Further, the main challenges related to energy efficiency, throughputs, latency, and packet loss ratio along scheduling processes have been outlined.

The comprehensive literature review of convergecast multichannel WSNs is given in Chapter 2. The motivation to carry out the present research work is also given in Chapter 2. Thereafter, the state of art work done by researchers is acknowledged and presented. The chapter includes motivation, a survey of convergecast WSNs, multichannel convergecast WSNs, energy utilization in WSNs, energy utilization in WSNs, and a comparative evaluation based on literature work. Further, the literature related to MAC protocols, multichannel routing, energy utilization, and scheduling methods has been discussed. Further, a comparative analysis of the techniques has been done in this chapter. Thereafter, the open issues in the current literature are pointed out and the objectives achieved in this work are elaborated on along with the research methodology.

Chapter 3 presents the proposed Fuzzy Based Energy Efficient Sleep Scheduling Mechanism (FBESSM) approach for convergecast WSNs. Firstly, the Framework of Convergecast in Multichannel Wireless Sensor Network is presented. Then, the traditional OSCAR technique utilized for WSNs is explained. Thereafter, the proposed FBESSM technique has been elaborated. Finally, the performance of this technique is calculated with different node counts and parameters such as packet delivery ratio, network lifetime, energy consumption and residual energy and compared with OSCAR algorithm. Chapter 4 presents the Efficient Multi Channel and Slot Allocation with Belief Score Authentication (EMCSABA) technique for slot allocation in convergecast WSNs. The architecture of the proposed scheme is elaborated. Evaluation of the EMCSABA based on varying node count and depth of node has been done under the network parameters such as PDR, energy consumption, network lifetime, and residual energy.

The Multichannel Light Weight MAC approach named MLMP is presented in Chapter 5. Firstly, the need for Multichannel in Convergecast WSN is described then the Implementation of Multichannel Convergecast in WSNs is explained.

Thereafter, the algorithm and working flowchart of the Multichannel Lightweight Mac Protocol (MLMP) technique are presented along with its comparative evaluation of network parameters such as packet delivery ratio, energy consumption, and residual energy. Chapter 6 discusses the conclusions from the research work and future directions.

CHAPTER 2

LITERATURE REVIEW

In a convergent network, each sensor node sends a packet straight to the sink node. Researchers find from these convergecast scenarios that for convergecast to delay economical and data reliable in emergency circumstances, real-time data collecting from individual nodes is necessary, as opposed to data aggregation scenarios. As a consequence, convergecast makes energy-efficient data gathering more difficult since each packet formed in a node must be transmitted to a sink node.

2.1 Motivation of Work

As described in Section 1.9, there are a lot of challenges associated with convergecast WSN such as energy, channel allocation, scheduling, and data aggregation. In multichannel WSNs, the sensor nodes can communicate with each other in a variety of ways. Different channels can be used to avoid interference and reduce collisions. However, using multiple channels also results in higher energy consumption because the nodes must switch between channels to talk to the sink [12]. Therefore, an efficient scheduling algorithm that minimizes energy consumption while ensuring reliable data transmission is necessary. The scheduling algorithm should also consider the delay requirements of the application, as delay sensitive applications need timely delivery of data to the sink. Taking these challenges into consideration, the aim of this thesis work is to develop or improve a scheduling algorithm for convergecast in a multichannel WSN which has higher energy efficiency as compared to existing algorithms [23].

Further, the topology of the network must be considered while designing a scheduling algorithm and it affects how well the WSN works. In multichannel WSNs, the topology can be dynamic, as the nodes can fail or fail to move, and the available channels can change. Therefore, an adaptive scheduling algorithm that can adjust to the changes in the network topology is required. Along with energy use, the scheduling algorithm should also take into consideration how long the network will last which depends on how much energy each node uses. The network lifetime is defined as the time until the first node runs out of energy. So, there needs to be a good scheduling algorithm that makes the most of the network’s life and makes sure data is sent reliably [25].

Now, the state of art literature related to multichannel convergecast WSNs is presented. For effective representation and to improve the clarity of the survey, the survey is structured into subsections. Section 2.2 presents the survey of traditional convergecast-based systems. The works related to multichannel convergecast WSN are presented in Section 2.3. The studies related to multi-channel MAC and multi-channel routing are given in Section 2.3.1 and 2.3.2 respectively. Further, Section 2.4 illustrates the literature on energy utilization in convergecast networks and Section 2.5 focuses on the data aggregation aspects. Thereafter, in Section 2.6, the scheduling techniques utilized for multi-channel WSNs are presented. Finally, the comparative evaluation based on the presented literature is carried out in Section 2.7

2.2 Survey of Convergecast WSNs

The radial coordination was suggested by Huang et al. in 2004 [30] as a new way to make sure that convergecast in WSNs works well. The approach is designed to simplify the process by which several sensor nodes provide data to a sink node and the sink node obtains this data, all by choosing a single parent node for each subtree in the WSN. Simulations are used to test how well the proposed approach works for energy usage and network lifetime. The work in [31] provides a new algorithm for convergecast in WSNs called LEACH-C, which is designed to reduce the latency and energy consumption of convergecast in WSNs. The algorithm uses a cluster-based method to coordinate the gathering of data from several sensor nodes and send it to a sink node.

The work by Upadhyayula et al. [32] proposes a new algorithm for Data Aggregation Convergecast (DAC) in WSNs called ST-DAC, which is based on a spanning tree approach to coordinate data aggregation and transmission from multiple sensor nodes to a sink node. The algorithm is designed to reduce the latency and energy consumption of DACs in

WSNs while ensuring reliable data transmission. The work in [33] talks about a way to cut down on the time it takes to collect data in WSN. The method makes a tree-like structure in the network and uses it to schedule data collection so that it doesn’t interfere with anything else.

Gandham et al. in 2006 [34] proposed a new scheduling algorithm for convergecast in WSNs called DMTCS (Distributed Minimal Time Convergecast Scheduling). The algorithm is made to reduce the volume of time it takes to collect and send data from multiple sensor nodes to a sink node while making sure that no two nodes conflict. Authors in [34] described the efficiency of proposed algorithm through simulations and show that it overtakes existing algorithms in terms of minimizing the time required for data collection and transmission. The authors in [35] propose a tree-based algorithm for converge casting in WSNs. The algorithm constructs a tree structure in the WSN and assigns schedules to nodes to ensure collision-free convergent broadcasting.

In 2008, Gandham et al. [36] presented a method for constructing a BFS tree for general network topologies, and the minimum time required to complete the convergecast is at most $3N-2$. However, this method uses a single channel, which can cause collisions between nodes in different branches. To avoid collisions, each node maintains a memory Table called a "conflicting map," which stores information about pairs of nodes that may interfere with each other. In another work [37], a convergecast scheduling algorithm using 2 channels has been proposed. Based on the topologies provided, the algorithm recommended a set of channels.

In [38], a technique called WirelessHART is proposed that allows the use of multiple packets and multiple channels. Unlike typical multi-channel TDMA protocols, WirelessHART uses channel hopping technology. In this technique, parallel transmissions occurring in a timeslot must select different channels to avoid collisions. In [39], a receiver-based frequency scheduling algorithm is proposed for multi-channel convergecast. The algorithm uses SINR to detect interference between parent nodes and assign non-conflicting channels to them.

Using directional antennas, such as Electronically Switched Directional (ESD) antennas, in WSNs may increase communication range and decrease interference with surrounding nodes by focusing transmission power in a specific direction. Tarter et al. in 2016 [40] explored the potential to use ESD antennas in WSN. As part of a convergecast application, they analyzed the performance of WSN protocols using quantitative methods. The study concludes that directional antennas do have some benefits, but they are limited and can only be used in certain situations. According to research conducted by Mottola et al. [41].

Further, Varshney et al. in 2014 [42] demonstrate that by making use of the capture effect and permitting several nodes to communicate to each other simultaneously, directional broadcasts and receptions may significantly decrease channel congestion. Wei et al. [43] demonstrate how a radio topographic imaging system that employs WSN nodes may benefit from the usage of directional antennas. In contrast, Varshney et al. [44] present a bulk transfer protocol that makes use of ESD antennas to send data via many, independent routes simultaneously, hence maximizing throughput.

2.3 Survey of Multichannel Convergecast WSNs

In this section, the convergecast by other researchers and scholars has been explored. The work is categorized into sub-sections about scheduling related, multichannel related, and energy using aggregation related concerning the convergecast in WSN.

2.3.1 Multichannel MAC

Interference and collisions are major challenges in WSNs, resulting in packet loss and reduced network throughput. To solve this problem, multichannel communication has been developed to prevent interference and collisions.

The minimal maintenance benefits of a fixed channel allocation are offset by the potential for worse performance in a dynamic operating environment for a WSN. Fixed channel allocation is used by both Tree-based Multichannel Protocol (TMCP) [45] and Multichannel Real-Time Communication MCRT [46] for communication and transmission which is used in multichannel communication. The Tree-based Multichannel Protocol (TMCP) is a Media Access Control (MAC) [45] protocol that uses tree partitioning to organize a physical network into many logical networks. All the nodes inside a given subtree have a single communication channel for exchanging information with one another. By designating distinct, non-adjacent channels, interference between adjoining sub-trees is prevented.

Within each sub-tree, the CSMA/CA algorithm controls medium access, and nodes communicate topology and routing information with one another. The results of GloMoSim calculations and testing with Micaz nodes show that TMCP increases throughput and decreases packet delay. Using a convergecast communication profile, MCRT [46] splits the network and assigns separate channels to each data flow to reduce channel congestion. The heuristic channel allocation method works to maximize the diversity of network pathways that converge on the target end-to-end communication latency. Mostly, these protocols aim to minimize data waiting times by doing away with inefficient and time-consuming channel switching.

Semi-dynamic channel allocation differs from static channel allocation in that each node is assigned a fixed channel but can also switch to available channels to communicate with neighbors. Hybrid Multi-Channel MAC (HMC-MAC) is a semi-dynamic multichannel MAC technique for WSNs proposed by Diabet al. in 2014 [47]. Each node in HMC- MAC chooses a reception channel within its immediate 3-hop neighborhood. The channel allocation method takes into account the nodes within each group. The node with the highest priority checks its immediate 3-hop neighborhood for free channels. To choose slots and channels, MC-LMAC [48], a MAC protocol based on TDMA, employs a decentralized approach. Each node has its dedicated communication window and channel. There are two distinct phases to the communication cycle: the control phase and the data transmission phase.

Multichannel adaptive allocation algorithms have been added to multichannel WSNs to increase resilience in the face of channel variance and traffic variations. Most adaptive allocation protocols, like Y-MAC [49] and MuChMAC [50], rely on precise synchronization sensors to get around the connection issue. The remaining channels are divided into two groups: high-priority and low-priority channels. To further save energy, the protocol employs a sleep/wakeup mechanism where nodes can sleep during idle times and wake up only when their timeslot arrives.

Many multichannel MAC protocols were included in the initial IEEE 802.15.4 standard as part of the IEEE 802.15.4e revisions so that the standard would better serve industrial applications. Time Slotted Channel Hopping (TSCH) and Deterministic and Synchronous Multi-channel Extension (DSME) are two such methods [51]. To accommodate commercial and industrial applications with demanding timeliness and reliability standards, the DSME protocol mixes contention based & TDMA and provides two channel diversity modes. The TSCH protocol makes use of multichannel communication techniques such as TDMA, channel hopping, and planned transmissions to mitigate the influence of interference and multipath fading. Although the TSCH protocol is a very effective energy-saving protocol, integrating it into the higher tiers of the communication stack may be challenging.

The protocol Adaptive (A-TSCH) is proposed by Du et al. in 2012 [52] as an enhancement to the TSCH protocol used in IEEE 802.15.4e. To adapt to changing wireless circumstances, A-TSCH employs a hardware-based channel energy measurement-based dynamic blacklisting mechanism to prioritize less-interference communication channels. In the work [53], the authors provided the Predictive Channel Scanning and Switching (PCSS) method. Its purpose is to lessen the amount of disturbance caused by wireless network nodes like Wi-Fi access points. The simulation findings show that PCSS is effective in lowering latency by decreasing the number of duplicated operations for channel switching and decreasing the average scan time needed to locate accessible channels.

There are use cases where the freshness of the supplied data is not optimal while switching interfaces often, making dynamic channel allocation undesirable. Many standards, such as MMSN [54] and TACA [55], are part of this movement. Given that it has been shown that loaded channels experience higher transmission time delays and more radio collisions, EMMAC [56] employs a node’s permutation channel schedule to shift among channels to minimize the load capacity of each channel. If the channel allocation is receiver-dependent, just like in EM-MAC [56], then independent hopping sequences need extra calculation and storage at every sender node. One of the frequency- key hopping's benefits is that it's less susceptible to interference and in case, the traffic load is high, RMCA [57] and ARM [58] algorithms use probability-based randomized channel selection.

To solve the scheduling, channel allocation, and power management problems simultaneously in multi-power WSNs, a new routing technique has been presented by Li et al. [59]. They came up with a heuristic strategy using the random walk technique to determine the best routes in to reduce both power usage and the end-to-end delay along pathways. Broadcast support becomes an issue that must be addressed when designing a multichannel MAC protocol. Current solutions to this problem involve introducing synchronization phases or using a signaling channel, which consumes time and energy and results in longer end-to-end delays. These solutions are also difficult to achieve in large-scale WSNs. Therefore, finding efficient solutions to support broadcast in multichannel MAC protocols remains an active area of research. The comparative analysis of these works has been depicted in Table 2.1.

2.3.2 Survey of Multi-Channel Routing

In a WSN with convergecast, all the data collected by the nodes is sent to the sink node, which is a central location. Following a routing metric, which may be static or dynamic, the routing protocol chooses the next-hop neighbor. Dynamic routing schemes, on the other hand, allow for the next-hop neighbor to shift during normal network operation based on the routing metric and the state of the network. In the hop count metric [60], each node in a WSN determines the route with the fewest hops to the sink. The static routing measure are inefficient and lead to congestion and data loss in WSNs with heavy traffic loads. Instead, dynamic routing improves with time due to the evolving value of the route measure.

To enhance data collection in multichannel WSNs, Chen et al. in 2014[61] proposed the Multi-Channel Collection (MCC) protocol which is a node-based time-scheduled routing protocol. The authors develop a Hybrid Multi-Channel (HMC) protocol with a tree- based routing strategy for WSNs in [62]. To enhance communication in densely interconnected wireless mesh networks, the authors of [63] propose the Joint Routing and Channel Assignment (JRCA) approach, which combines routing and channel assignment. JRCA employs a link-based routing measure that balances the probability of successful communication with the time it takes to deliver a packet to determine the optimal multi-hop route.

Table 2.1: Comparative evaluation of channel allocation techniques on WSN

Ref.	Static Channel	Dynamic Channel	Semi Dynamic Channel	Robust	Delay Efficient	Energy Efficient	QoS	Scalability
[45]	Yes	No	No	No	Yes	No	No	No
[46]	Yes	No	No	No	Yes	No	Yes	No
[54]	No	No	Yes	No	No	No	No	No
[55]	No	No	Yes	Yes	No	No	Yes	No
[56]	No	No	Yes	Yes	No	No	No	No
[89]	No	No	Yes	Yes	Yes	Yes	Yes	No
[57]	No	No	Yes	Yes	No	No	Yes	No
[58]	No	Yes	No	No	Yes	Yes	No	Yes
[49]	No	Yes	No	No	No	Yes	No	Yes
[50]	No	Yes	No	No	No	Yes	No	No

The literature on channel allocation techniques in WSNs as shown in table 2.1 presents conflicting findings. Static channel allocation is commonly associated with delay efficiency, whereas dynamic approaches are often associated with energy efficiency and scalability. Semi-dynamic methods show varied support for robustness and quality of service (QoS). Notably, comprehensive evaluations addressing all key metrics simultaneously are scarce, highlighting divergent research focuses and application priorities.

Real-time communication via WSNs is specifically addressed by the Multi-Channel Real-Time (MCRT) protocol introduced in [64]. Using multichannel data transmission, this unified MAC and routing system can keep the total network latency of all data flows under a predetermined limit. To locate and eliminate congestion in WSNs, the Collision Avoidance and Detection (CODA) routing protocol is defined in [65]. To detect network congestion, this protocol employs a pair of techniques. One drawback of this protocol is that it prevents event-detecting nodes from sampling and transmitting data that would otherwise be received by the sink node.

Data aggregation techniques can be used to take advantage of spatial or temporal correlations in sensed data to reduce the amount of data and keep the network from getting clogged up, especially in WSNs that collect data [66]. An adaptive data aggregation-based congestion control protocol called CONCERT is proposed in [67]. The protocol leverages the high degree of spatial correlation in data collected by sensor nodes, where multiple nodes in the same area collect and transmit the same information in the case of a specific event. A comparison of various approaches in convergecast WSNs is presented in Table 2.1.

A protocol was proposed by Junior et al. in 2014 [68] named CodeDrip to disseminate small values in WSNs. The key idea at the back of this set of rules is to employ Network Coding to reduce the number of transmitted packets and to save energy. Multihop Over-the-Air Programming (MOAP) protocol was proposed in [69] which is a reprogramming protocol. This protocol addressed the problems of XNP. MOAP disseminates code and modernizes it into a multi-hop communication fashion. The authors of [70] proposed a centralized energy-efficient collision-free scheduling algorithm.

Further, in [71] TreeMAC algorithm is proposed where a TDMA based MAC protocol is considered for real time high data rate applications. The authors of [72] presented another protocol to assign diverse channels to vertex-disjoint trees which are rooted at the sink and take advantage of parallel data transmissions amongst trees. Further, the authors of [73], proposed two algorithms to schedule data aggregation in multi-sink wireless sensor networks. A load-balancing problem was pointed out by [74]. To handle this issue novel convergecast tree protocol was proposed along with the distributed adjustment algorithm which attained load balancing and also expanded the lifespan of the network.

Further, to resolve the data discarding issue of adhoc WSNs, ALBA algorithm has been proposed in [75]. This approach is comprised of Adaptive Load Balancing Algorithm (ALBA) along with priority. The authors of [76] modified the aggregation convergecast model, data collection delay and decoupling schedule length by relaxing its precedence constraints. To equilize the delay time and encoding number, a Converge-cast Scheme based on data collection rate prediction (CSRP) was proposed by Xu et al. in 2017 [77].

A survey on end-to-end encryption clarification for convergecast data traffic in WSNs is presented in [78]. Shiet al. in 2019 [79] studied the issue of LACSs-based convergecast scheduling in industrial WSNs incorporating tree topologies. A novel method based on a path duration model for convergecast data aggregation has been proposed by [80]. An optimum convergecast scheduling for a wireless sensor network has been proposed by Bakshi in 2017 [81]. The author provided a mathematical formulation for the hard NP-hard problem. Another work in the direction of multichannel convergecast in WSN has been reported in [82]. The work is focused on collision free schedules of raw data with minimum delay within multi-sink multichannel WSNs. The authors extended the work and reported the outcomes in [83]. In this work, the authors studied the raw data convergecast of mutlichannel WSNs for heterogeneous traffic. Recently, the power- efficient and interference free link scheduling problem has been addressed by [84].

2.4 Survey of Energy Utilization in WSNs

Numerous real-time control messages are required to keep an established route due to the poor dependability of the sensor network and wireless communication. These activities need a lot of additional bandwidth and energy. Thus, studying the role of energy is an important concern. A Systematic Data Aggregation Model (CSDAM) for processing data

Table 2.2: Comparison of various approaches in Convergecast WSNs

Ref.	**Approach/ algorithm**	**Contribution**	**Performance matrices**	**Drawbacks**	**Application**
[78]	CodeDrip	Employment of Network Coding to reduce the amount of transmitted packets and to save energy	Recuperating lost packets	Limited parameters evaluated	data dissemination
[69]	Multihop Over-the-Air Programming (MOAP)	Disseminate code modernize it into a multi-hop communication Fashion	Dissemination of code by neighborhood-by-neighborhood	Neighborhood- by-neighborhood fashion only	data dissemination
[70]	Centralized Collision-free Scheduling	Set of trees in going forward	diverse rounds and load on sensor nodes	Limited topology	convergecast scheduling
[71]	TreeMAC	Estimation of possible horizontal and vertical collisions	Throughput	Limited to same path from the root	convergecast scheduling
[72]	Vertex-Disjoint Trees	Assignment of diverse channels to vertex-disjoint Trees	Throughput and collisions	Data aggregation to a single sink.	convergecast scheduling
[73].	Convergecast Tree Protocol	Load balancing	Lifespan of the Network	Distributed adjustment	convergecast scheduling
[74]	ALBA and ALB-P	The data discarding issue of adhoce WSNs	Latency	Adaptive Load Balancing	convergecast scheduling
[75]	Modified Aggregation Convergecast	Decoupling schedule length, and data collection delay	Latency	snapshot pipelining	convergecast scheduling
[76]	Data collection rate prediction	Increased the coding prospective and avoided the collision problems	Collision	Limited real time scenario	convergecast scheduling
[79]	LACSs-based Convergecast Scheduling	Formulation developed for programming in convergecast scheduling.	Latency	Lack of construction of tree routing paths	multichannel convergecast scheduling
[80]	Convergecast Data Aggregation	Novel method based on path duration	Latency and throughput	Limited employment multiple access technologies	convergecast scheduling
[81]	Optimum Convergecasts Scheduling	Mathematical formulation for the hard NP- hard problem	Latency and throughput	Difficult to integrate the channel impairment	multi-sink multichannel convergecast scheduling
[82]	DiSCA	The raw data convergecast of mutli-channel WSNs for heterogeneous Traffic	Latency and throughput	Does not include the all the interfacing links but supports scheduling	multichannel convergecast scheduling
[83]	GRASP-based Metaheuristics	Comparison of power consumption and schedule length	Power consumption and schedule length	Limited data dissemination and communication pattern	convergecast scheduling
[84]	CHOPIN	Makes the network much more adaptable to changing external Conditions	Convergence time for energy	Limited scalability and global message flooding	convergecast scheduling

in real-time is presented in [85]. The network is organized into a cluster, consisting of both awake and inactive nodes, and a Cluster-Head (CH) is chosen based on the sensor rankings in terms of their current energy status and their distance from the Base Station (BS). As discussed by the authors of [86], data aggregation is a useful method for extending the useful life of WSNs and decreasing their energy consumption.

Each sensor node, as described by the authors of [87], looks for other sensor nodes that have made identical readings within a certain period and aggregates them. To improve energy conservation and extend network life, the researchers in [88] have conducted thorough research on this type of interconnected sensor-based network. In [89], the authors suggest an energy-efficient data-gathering format for use in WSNs. The sensors are gathered for probable rendezvous location selection using clustering techniques. In [90], the authors suggest a pattern-based Redundancy Elimination Data Aggregation (REDA) technique they term. The suggested pattern is data-driven and makes use of differential data amassed over several iterations at distinct sensor nodes.

Adaptive Data Collection (ADC) is a methodology proposed by the authors of [91] for collecting sensor data at regular intervals to increase the life of a Periodic Sensor Network (PSN). The ADC protocol lifespan is quantified in cycles. There are two phases to every cycle. The authors offer multi-layer large data aggregation architecture and a Priority-Based Dynamic Data Aggregation (PDDA) approach [92]. Since most current methods focus only on data aggregation at the central server level, the suggested PDDA methodology operates at the bottom layer of sensors.

In [93], an effective clustering technique called the Low-Energy Adaptive Clustering Hierarchy (LEACH) is explained; in which nodes inside a cluster submit their information to a local CH. In [94], the authors offer a protocol called Hybrid Energy-Efficient Distributed Clustering that chooses CHs periodically based on a combination of the node’s remaining energy and a supplementary variable, such as the node’s closeness to its peers or its grade. It provides a fairly consistent distribution of CHs over the network with little message overhead.

Utilizing fuzzy logic, the distributed clustering method Cluster Head Election (CHEF) is presented in [95] which can function with WSNs. One such fuzzy-based technique for WSNs is the energy-aware, distributed dynamic clustering protocol using fuzzy logic (ECPF) [96]. There are two stages to the protocol: initialization and maintenance of the current state. At this first stage, the CH is elected, and clusters develop. During the steady-state period, TDMA frames are generated, and data is collected. The fuzzy cost is the result of ECPF's use of the input variables node degree and node centrality. Each node in the network experiences a slowdown which is proportionate to its residual energy in ECPF.

Specifically developed for wireless sensor networks, the Fuzzy Inference System (FIS) provides the basis for the computation of the competition radius of candidate cluster heads in the Multi Objective Fuzzy Clustering Algorithm (MOFCA) as presented by Sert et al. in 2015[97]. While making its calculations, the FIS includes factors such as residual energy, distance to sink, and node density. TSCH is defined under the IEEE 802.15.4- 2015 standard [97]. It is the fastest and most efficient option, using the least amount of energy. Congestion is particularly bad for convergecast nodes close to the root. This issue is addressed in OSCAR [98] by Osman et al. which is a revolutionary independent scheduling TSCH cell assignment technique built on Orchestra.

OSCAR, in contrast to Orchestra, uses the node's distance from the root to determine how many slots each node receives. The main goal of the study by Ghosh et al. conducted in 2022 [99] is to find the best way to gather and combine data in WSNs by using multiple frequency channel assignments. Due to how networks are set up and how interference works, using multiple frequencies for communication is more effective than using a single frequency. John et al. in 2021 [100] came up with a way for wireless sensor networks to collect data for as long as possible. They discovered that the energy consumption of two nodes is the same if they deliver the same amount of data packets, regardless of the length of the payloads within those packets. The goal of the work in [101] is to reduce the cost of transmitting collected data in remote monitoring systems while maintaining data throughput requirements.

Researchers have proposed a new algorithm, the multi-objective Adam hybridized flower pollination optimization algorithm to find the most efficient way to send data from the source to the destination while taking into account a variety of factors, including throughput, service cost, and energy consumption. A variety of techniques and strategies have been suggested to improve the primary performance aspects of WSN MAC protocols, and they are discussed by Sadeq et al. in 2022 [102]. A comparative chart is presented to illustrate how these methods and algorithms may enhance network throughput, end-to-end latency, packet loss, and energy usage. The MAC protocol should handle these numerous performance problem variables.

2.5 Survey of Data Aggregation

By aggregating data from multiple nodes before transmitting it to the sinks, the amount of data traffic and energy consumption can be significantly reduced, thereby extending the network’s lifetime. This can help to increase the available bandwidth and reduce the delay and packet loss in the network [103-104]. But the problem is how to give each node the right amount of time slots and channels. This is because the nodes need to talk to each other to make sure they don’t send the data at the same time and on a similar channel, which can cause collisions and packets to be lost as discussed in a study by Huang et al. [105]. Different scheduling algorithms, such as TDMA FDMA, and CDMA have been proposed to solve this problem. In some of the works, data from throughout the network is gathered, and optimum scheduling output is computed [106-107].

Kumar et al. [108] present a multi-channel TDMA scheduling technique that prioritizes low-power network operation. The method employs several Radio Frequency (RF) channels to allow simultaneous conversations and minimize the likelihood of collisions. In addition, the continuous link scheduling issue in WSN is explored by Ma et al. [109]. This method schedules tasks for each node throughout a continuous period, allowing each node to sleep for long periods before being roused to do its work. Further, Yao et al. in 2021 [110] were the first to study the many-to-many communication scheduling issue for WSNs that did not rely on batteries. In order to schedule the collection of data from numerous sensors in a multi-channel WSN, Bagaa et al. [111] suggested a trusted data aggregation scheduling mechanism that works across several network levels. Similarly, Jiao et al. [112] demonstrated that the data aggregation scheduling issue for multi channel duty cycle wireless sensor networks is hard. In order to solve this problem, they adopted a coloring approach to depict the scheduling interactions between nodes using candidate-activity conflict and feasible-activity conflict graphs.

For multi-channel WSNs, many researchers have developed distributed data aggregation scheduling techniques. In order to resolve the issue of scheduling in duty cycle wireless sensor networks, Kang et al. [113] developed a distributed delay effective approach. In order to better schedule many-to-one applications, Lu et al. included a separate Q-learning method into an adaptive time slot scheduling framework. When frames are transmitted, this scheduling mechanism approaches the best possible outcome [114]. In [115], Ren et al. introduced a cluster-based distributed data aggregation scheduling method that takes into interpretation both the available power and the number of available channels. The comparative analysis of the literature presented in this section is depicted in Table 2.3.

2.6 Survey of Scheduling Techniques

In this section, the field of convergecast related to the scheduling algorithms in WSN is presented. Some of the most relevant works have been discussed here. In centralized systems, network data is sent to a central organization, which then schedules connections. In Traffic Aware Scheduling Algorithm (TASA), the schedule is made by a single node as studied by Palattella et al. [116] where in the network's layout and current traffic levels are taken into account to determine the schedule followed by each node. To reduce idle flow, Hoi et al. in 2016 proposed Cooperative Link Scheduling (CLS) which distributes and discharges slots without altering the whole schedule at every CLS [117]. This results in effective multi-hop schedules with a minimal number of centralized control messages.

Multichannel Optimized Delay Time Slot Assignment (MODESA) was first presented in [118]. It is a centralized raw convergecast type network. DeTAS [119] deals with networks that have a lot of sink nodes. In [120], the authors suggest a method for scheduling that is based on waves. Each sending node is allotted a certain amount of time in a specific channel for sending its packets during each wave. Aijaz et al. in 2017 [121] proposed DeAMON which is a decentralized mechanism for 6TiSCH wireless networks. Later, a distributed diverge cast scheduling method called DIVA [122] was proposed.

Autonomous scheduling is better for the network than centralized and distributed scheduling because it uses less bandwidth. Duquennoy et al. in 2015 proposed Orchestra which began auto-scheduling TSCH [123]. Orchestra is a new way for TSCH to schedule jobs that uses basic criteria to speed up the rate of delivery. Further, the Escalator [124] is an autonomous scheduling method that uses Orchestra. It makes a sequential timeslot plan along the packet transmission pipeline to cut down on latency. Rekik et al. in 2018 [125] tried to cut down on orchestra delays by changing the order in which they played. The authors of [126] propose a deterministic TDMA scheduling technique for dispersed networks. FlexiTP, presented in [127], is a unique TDMA protocol that provides a synchronized and flexible slot structure. A distributed and scalable scheduling accessing approach was presented in [128].

For solving the problem of communication resource allocation in a tree topology, Mei et al. [129] present a matching scheduling algorithm (MSA). In order to reduce the excessive latency and the growing number of channels caused by an expanding network, the inventors of MSA have improved the algorithm. The algorithm is known as OSA (optimized scheduling algorithm). With the multi-channel capabilities of the radios fully used, Ha et al. [130] suggest a novel technique that increases network throughput. To improve the mechanism in the Routing Protocol for Low-Power and Lossy Networks (RPL) and TSCH MAC layer, Georgios et al. [131] suggested a leap frog collaboration. In order to duplicate its data, each node chooses two parents, an AP parent and a DP parent. A unicast data packet is sent from each child to each parent. Developed by Seohyang et al. in 2021 [132], ALICE is an autonomous link-based cell scheduling used in RPL routing. The goal of ALICE is to schedule cells with no extra overhead, block interference between upstream and downstream, and maximize the use of all available channels.

Collision Free Advertising Scheduling (CFAS) was suggested by Apostoloset al. [133] as an autonomous Improved Beacon scheduling strategy that prevents collisions. Using this method does not result in a rise in required energy input but also improve the other associated factors. Without requiring any kind of interaction between nodes, CFAS generates EB scheduling on a multi-slot frame structure. CFAS states that the quantity of available ad spots is proportional to the total number of advertisers and the targeted EB rate. Tavallaie et al. in 2021[134] concentrated in on wireless systems and an innovative Distributed Traffic-aware Scheduling Function (DT-SF) has been suggested which allows the TSCH schedule to be flexible in response to shifts in network topology and traffic demands. Some other latest works pertaining to convergecast has been done by various researchers [135-138].

2.7 Comparative Evaluation Based on Literature Work

In this section, a comparison of the different ways that convergecast is done in WSN has been made. Channel allocation is one of the prime concerns that have been noticed through the literary work which also embarks on the efficiency of any convergecast approach. In Table 2.2, various methods for channel allocation have been evaluated based on their type of allocation of the channel along with their Quality of Service (QoS) parameters such as robustness, delay, energy, and scalability. The three broad categories for channel assignment approaches: static, semi-dynamic, and dynamic are presented in Table 2.2. Approaches that do not evolve are static which limits their usefulness. This means they are useful in situations when both the surroundings and the application are stable and well-understood. The semi-dynamic strategy is appropriate when conditions and traffic levels remain constant over an extended period of time.

Table 2.3 demonstrates that there are three distinct categories of aggregating algorithms: centralized, decentralized, and in-network. In a centralized approach, a central node either acts as the data aggregator and coordinator itself or appoints another node to do so. In the in-network approach, the algorithm operates locally and determines both the data aggregator and the aggregation method. Initial research often believed that any of the nodes may serve as the aggregator even though this was not the case. Since then, methods have been introduced in which aggregators are mobile nodes that are not limited by available resources, allowing them to freely roam the network and gather information from a variety of nodes.

In static networks, where no change is observed in the architecture over time, the cluster-based technique is quite effective. However, such a method is usually ineffective when used inside a dynamic network. Hence, it would be beneficial to investigate data aggregation strategies for wireless or adap Table networks. Table 2.3 further suggests that the primary inclination is toward the dispersed nature and that most work has been done on homogeneous nodes rather than heterogeneous nodes. In addition, data aggregation using a tree structure has received less attention. Convergecast is divided into three categories: aggregate, raw, and general. Also, the energy used, the length of the schedule, and the ability to handle problems are the prime factors to look at when evaluating how well the different convergecast techniques work. In this direction, Table 2.4 shows a comparison of different convergecast techniques proposed in the literature.

Table 2.3: Comparative evaluation of data aggregation techniques on WSN

Ref	Cluster Based	Tree Based	Homogenous Node	Heterogeneous Node	Mobility	Centralized	Distributed	In-Network
[85]	Yes	No	Yes	No	No	No	Yes	No
[86]	Yes	No	Yes	Yes	No	No	Yes	No
[87]	Yes	No	Yes	No	No	No	No	Yes
[88]	Yes	No	Yes	No	Yes	No	Yes	No
[89]	No	No	Yes	No	Yes	No	Yes	No
[90]	Yes	No	Yes	No	No	No	Yes	No
[91]	Yes	No	Yes	No	No	No	Yes	No
[92]	Yes	Yes	Yes	No	No	No	Yes	No
[93]	Yes	No	Yes	No	Yes	No	Yes	No
[94]	Yes	No	No	Yes	No	No	Yes	No

Based on the scheduling algorithms and fault-tolerance methods listed above, it can be concluded that joint scheduling or general convergecast algorithms are better than other methods. The length of the schedule is short in a small network whereas the length of schedules is long in a big network. If something happens in one part of the network, the nodes in a big network will have to wait longer for their turn to send. So, for faster data transfer, researchers need to come up with methods that use less energy and can handle mistakes and have the shortest schedules.

There are a lot of various viewpoints in the studies depicted from table 2.3 in WSNs. There are some studies that use tree-based techniques and different kinds of nodes, but most of them use cluster-based methods and models with homogeneous nodes. Mobility support isn't common, but some works take mobile nodes into account. There are almost no centralized methods. Instead, distributed techniques are used a lot. In-network processing is also only talked about in one study, while external grouping methods are talked about in many more. These differences show that WSN research uses a wide range of methods and situations, which is because different applications have different needs and study priorities.

Table 2.4: Metric based evaluation of convergecast techniques

Reference	Aggregate Convergecast	Raw Convergecast	General Convergecast	Energy	Schedule Length	Robustness
[126]	No	No	Yes	No	No	No
[127]	No	Yes	No	No	No	No
[87]	Yes	No	No	Yes	No	No
[128]	No	No	Yes	No	No	No
[89]	Yes	No	No	Yes	No	No
[91]	No	Yes	No	Yes	No	No
[92]	No	Yes	No	No	Yes	No

There is an abundance of various views in the research on convergecast methods in WSNs as depicted from table 2.4. Some studies focus on bulk convergecast and emphasize how energy-efficient it is, while others focus on raw convergecast and often lack concern about energy. Some researchers discuss general convergecast, but they don't look at energy or schedule length. Different studies use different methods to measure energy metrics, and some studies don't even look at them at all. Only one study investigated it at schedule length, which means it's not often thought about. None of the works that were looked at test robustness. These differences show the different ways of doing things and the different priorities in convergecast research, which is a reflection of the different needs and goals of the study.

2.8 Objectives of the Proposed Work

Based on the literature review presented in Chapter 2, the following research objectives are formulated for the work presented in this dissertation:

- To develop energy utilization algorithm for improving the energy efficiency of convergecast multichannel wireless sensor network.
- Develop a scheduling algorithm to improve reliability of wireless sensor network by minimizing collisions.
- Propose and develop multichannel communication architecture for wireless sensor network.
- Performance evaluation of proposed algorithm with existing techniques.

2.9 Research Methodology

Initially, a thorough literature survey is carried out about the convergecast multichannel wireless sensor network. Afterwards, a convergecast in multichannel wireless sensor network topology is created with multiple nodes. The WSN is implemented based on IEEE 802.15.4 standard to transmit the packets from source to destination. This is followed by developing an energy utilization algorithm to increase the energy factors of the convergecast multichannel wireless sensor network. The proposed approach is named Fuzzy Based Energy-Efficient Sleep Scheduling Mechanism (FBESSM) where fuzzy logic is integrated for the CH selection based on the sleep scheduling approach for efficient energy utilization. A comparative analysis of the proposed FBESSM is conducted with OSCAR algorithm for different performance metrics like throughput, network lifetime, end-to-end delay, energy consumption, residual energy, and packet loss with varying node counts ranging from 50 to 250 have been explored to get insight into their performance. In a subsequent step, an improved multichannel architecture for wireless sensor networks is proposed named Efficient Multi-Channel and Slot Allocation with Belief Score Authentication (EMCSABA). Incorporating a belief score authentication scheme strengthens the security of data aggregation, allowing only trusted nodes to participate in the convergecast process. Performance indicators like as throughput, packet loss, lifespan, energy consumption, and latency are being used to assess the effectiveness of the same with other methods like FBESSM and OSCAR. Also, the proposed algorithms are evaluated and compared with existing multichannel wireless sensor networks based on their performance through a new approach named MLMP which is an efficient way of allocating channels in the multichannel environment for convergecast WSN. The proposed projected algorithm allows nodes in a wireless sensor network to autonomously select non-conflicting timeslots to transmit their data, ensuring communication without interference.

Using algorithms like FBESSM, EMCSABA, and MLMP in the real world is not easy for a number of reasons. For FBESSM, it can be hard to keep energy efficiency high when network conditions are uncertain and make sure fuzzy logic decisions are reliable. Because EMCSABA uses multiple channels and slots, it needs to be perfectly in sync and have strong belief score authentication, which can use a lot of resources. MLMP's multichannel management has to deal with interference and changes in the network's structure on its own. All of these algorithms also need to be able to handle a lot of nodes and have to find a balance between additional work and limited energy. Making sure they work well with current systems and equipment adds another level of challenge to their effectiveness in the real world.

CHAPTER 3

FBESSM: A FUZZY BASED ENERGY EFFICIENT SLEEP SCHEDULING MECHANISM

Energy efficiency criteria for convergecast WSN networks include decreasing energy consumption during data transmission from numerous sensors to a central node. By preserving the energy resources of sensor nodes, methods including data aggregation, and adaptive routing. The wireless sensor networks of today are designed to collect and process vast quantity of data. Congestion on the network and complicated procedures caused less efficiency even in the case of OSCAR technique. Because of the proximity of the nodes, collisions during transmission and energy consumption from duplicate data will be unavoidable. To address these issues, in this chapter a new approach is presented which is a fuzzy-based sleep scheduling mechanism (FBESSM) that would activate, or sleep sensors as needed to decrease power usage. Performance indicators like throughput, packet loss, lifespan, energy consumption, and latency are being used to assess the effectiveness of the FBESSM approach as it has been implemented in NS-2.

3.1 Framework of Convergecast in Multichannel Wireless Sensor Network

Creating a Convergecast in a Multichannel Wireless Sensor Network (WSN) with multiple nodes involves several steps to implement the IEEE 802.15.4 standard algorithm. In this section, we have discussed the process in detail.

Step 1: Network Topology Design

The first step in creating a Convergecast in a Multichannel WSN is to design the network topology. The network topology refers to the physical layout of the nodes in the network. A multichannel WSN can have various topologies, such as star, mesh, and tree.

Step 2: Channel Allocation

After designing the network topology, the next step is to allocate channels to the sensor nodes. The IEEE 802.15.4 standard algorithm defines 16 channels (from 11 to 26) for the operation of WSNs. The channel allocation algorithm assigns different channels to different sensor nodes to avoid interference and congestion on a particular channel. In multichannel WSNs, the channel allocation algorithm should also take neighboring networks interference into account.

Step 3: Routing

Routing is the procedure of choosing the optimal path for transmitting the packets from the source to the destination. In a tree topology, the routing algorithm should select the shortest path from the source node to the sink node. The IEEE 802.15.4 standard algorithm provides two types of routing algorithms: Mesh Under and Mesh Over. The Mesh Under routing algorithm is suitable for tree topologies, as it confirms delivery of the packets to the sink node without looping back.

Step 4: Data Aggregation

Data Aggregation is the process of merging multiple data packets into a single packet before transmitting it to the further source. Data aggregation at intermediate nodes reduces the amount of data transmitted and improves the efficiency of the network. The IEEE 802.15.4 standard algorithm provides a Data Aggregation function that enables data aggregation at intermediate nodes.

Step 5: Channel Switching

Channel Switching is the process of switching channels when the channel becomes congested or noisy. In multichannel WSNs, channel switching is critical to avoiding interference and ensuring reliable communication. The IEEE 802.15.4 standard algorithm provides a Channel Hopping function that enables the sensor nodes to switch channels periodically.

Step 6: Quality of Service (QoS) Management

QoS management is the process of managing the network resources and channel access to meet the QoS requirements of the application. The IEEE 802.15.4 standard algorithm provides a Guaranteed Time Slot (GTS) function that enables the nodes to allocate time slots for high-priority data transmission.

Step 7: Energy Efficiency

Energy efficiency is critical in WSNs, as the sensor nodes are typically battery-powered and have very less energy resources. The IEEE 802.15.4 standard algorithm provides several energy-efficient protocols, such as the Low-Rate Wireless Personal Area Network (LR-WPAN) protocol and the Battery Life Extension (BLE) protocol.

Step 8: Network Monitoring

Finally, the network should be monitored for faults, congestion, and other performance issues, and appropriate measures should be taken to ensure smooth network operation. The IEEE 802.15.4 standard algorithm provides a Network Management function that enables network monitoring and management.

3.1.1 IEEE 802. 15.4 Protocol

IEEE 802.15.4 is a standard for low-power wireless personal area networks designed for short-range, low-data-rate communication between devices. The standard provides specifications for the physical layer (PHY) and medium access control (MAC) layers, channel access methods, frame formats, security, network topologies, device discovery and association, and quality of service (QoS) management.

The slotted CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) algorithm is used in the IEEE 802.15.4 standard for medium access control (MAC) layer communication. It is a contention-based protocol that allows multiple nodes to share the same channel for transmitting data.

The working of the slotted CSMA/CA algorithm can be explained as follows:

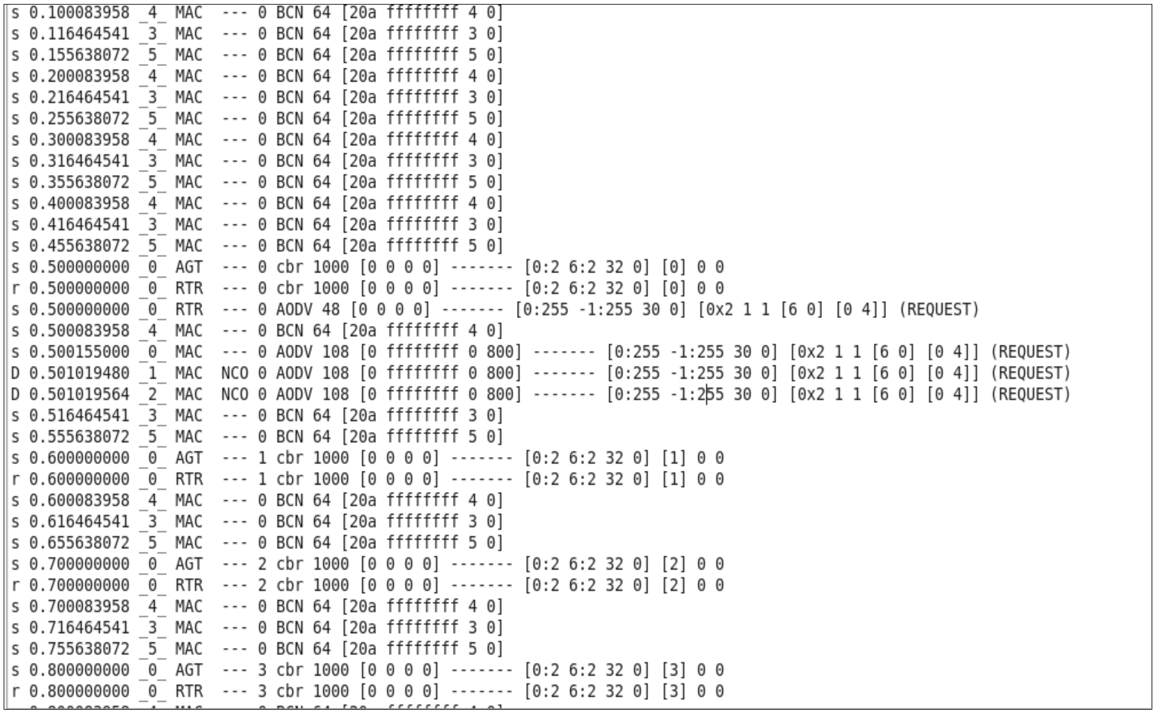
- Time is divided into a series of fixed-length slots. Each slot has a predefined length, typically 16 or 32 microseconds.
- Each node listens to the channel to detect any ongoing transmission. If the channel is busy, the node waits until the channel becomes free for transmission.

- If the channel is idle, the node waits for some time, known as the back-off time, before attempting to transmit. The back-off time is calculated based on a contention window (CW), which determines the maximum number of timeslots that the node must wait before it can attempt to transmit.
- Once the back-off time expires, the node checks the channel again to ensure that it is still idle. If the channel is busy, the node restarts the back-off timer and waits again.
- If the channel is still idle, the node begins transmitting its data. It transmits data in a single slot, or it can spread the data over multiple slots if the data size is larger than a single slot.
- After transmitting its data, the node waits for an acknowledgement (ACK) from the receiver. If the node receives an ACK, it considers the transmission successful. If it does not receive an ACK, it assumes that the transmission was unsuccessful due to a collision or other interference, and it retries the transmission after waiting for a random backoff time.

The slotted CSMA/CA algorithm ensures that nodes avoid collisions and interference by using a random back off time and a contention window. By dividing time into fixed- length slots, the algorithm allows nodes to access the channel in a distributed manner, ensuring fairness and efficiency. The working process of IEEE 802.15.4 protocol has been represented in the form of a flowchart as shown in Fig. 3.1.

The slotted CSMA/CA algorithm for IEEE 802.15.4 protocol follows a set of steps for transmitting frames. These steps include transmitting frames, performing clear channel assessments (CCAs), and selecting back-off duration. The back-off duration is increased with each attempt until it reaches a maximum value. If the number of back-offs reaches its maximum limit, a transmission failure is reported to higher layers. After delaying for a random duration, the MAC sub layer proceeds if it can complete two CCAs, data frame transmission, and ACK frame reception within the time slot. If not, the operation is deferred to the next time slot. The transmitting node listens to the common channel during each CCA for eight periods. If the channel is sensed as idle during the two successive CCAs, transmission of a frame begins. The contention window (CW) counter value is decremented by 1 if the channel is sensed idle during a CCA, and reset to 2 if either of the two CCAs fails. The second CCA is needed to prevent collisions with ACK frames.

The details of the log created while implementing the Convergecast have been expressed in Fig 3.2.The NS-2 log files as shown in Fig. 3.2 offer comprehensive details about many metrics and events across network simulations. The log files typically contain information about packet transmission/reception timestamps, route establishment, routing Table updates, MAC layer interactions, link quality, packet drops, and protocol-specific messages for metrics like Media Access Control (MAC), Beacon (BCN), Constant Bit Rate (CBR), and Ad hoc On-Demand Distance Vector (AODV) protocol. These logs are useful for troubleshooting problems in the simulated environment and analyzing protocol performance and network activity.



```
s 0.100083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.116464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.155638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.200083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.216464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.255638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.300083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.316464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.355638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.400083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.416464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.455638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.500000000 _0_ AGT  --- 0 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [0] 0 0
r 0.500000000 _0_ RTR  --- 0 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [0] 0 0
s 0.500000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
s 0.500083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.500155000 _0_ MAC  --- 0 AODV 108 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
D 0.501019480 _1_ MAC  NCO 0 AODV 108 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
D 0.501019564 _2_ MAC  NCO 0 AODV 108 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
s 0.516464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.555638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.600000000 _0_ AGT  --- 1 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [1] 0 0
r 0.600000000 _0_ RTR  --- 1 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [1] 0 0
s 0.600083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.616464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.655638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.700000000 _0_ AGT  --- 2 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [2] 0 0
r 0.700000000 _0_ RTR  --- 2 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [2] 0 0
s 0.700083958 _4_ MAC  --- 0 BCN 64 [20a ffffffff 4 0]
s 0.716464541 _3_ MAC  --- 0 BCN 64 [20a ffffffff 3 0]
s 0.755638072 _5_ MAC  --- 0 BCN 64 [20a ffffffff 5 0]
s 0.800000000 _0_ AGT  --- 3 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [3] 0 0
r 0.800000000 _0_ RTR  --- 3 cbr 1000 [0 0 0 0] ------- [0:2 6:2 32 0] [3] 0 0
```

Fig 3.1: Details of a log created for packet transmission/reception timestamps, route establishment, and routing table updates.

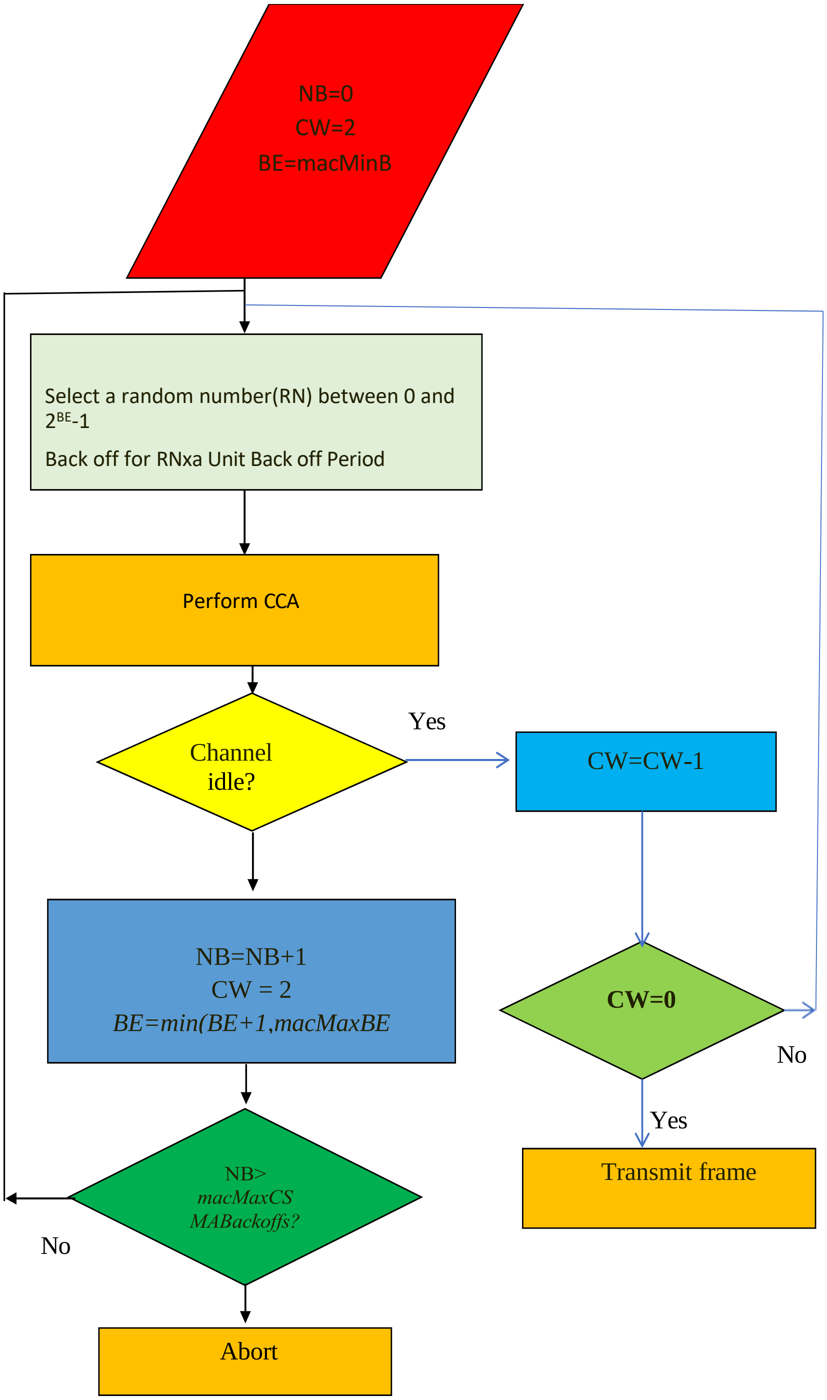
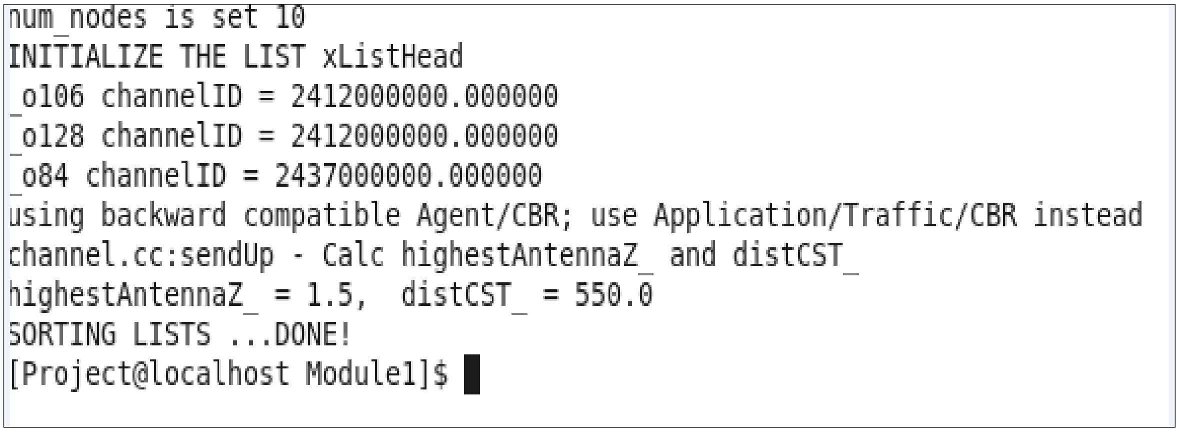


Fig. 3.2: IEEE 802.15.4 protocol

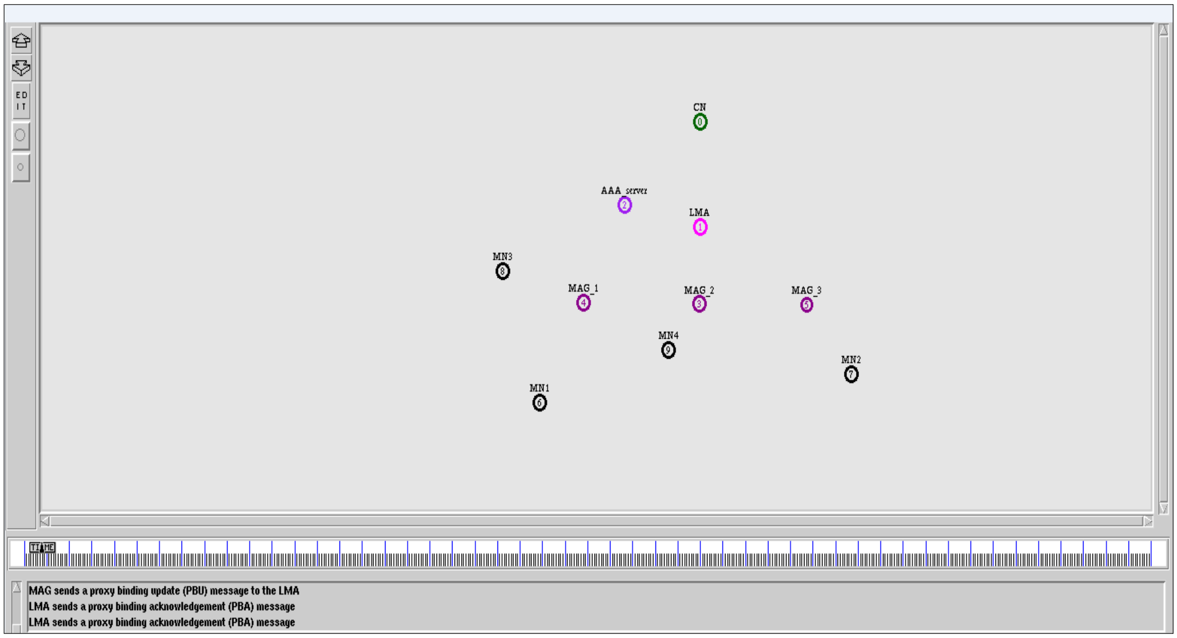
All of these parameters help to make convergecast activities in WSNs effective by guaranteeing dependable communication, efficient energy usage, reduced collisions, and prompt data delivery to the central node for processing.



```
num_nodes is set 10
INITIALIZE THE LIST xListHead
_o106 channelID = 2412000000.000000
_o128 channelID = 2412000000.000000
_o84 channelID = 2437000000.000000
using backward compatible Agent/CBR; use Application/Traffic/CBR instead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
[Project@localhost Module1]$ █
```

Fig 3.3: Multichannel selection for Convergecast

Fig 3.3 highlights the multiple channels that are being used for the Convergecast as it has been visible that three channel IDs are being generated. Fig 3.4 showcases the visualization of the proposed multichannel network that is being deployed.



```
MAG sends a proxy binding update (PBU) message to the LMA
LMA sends a proxy binding acknowledgement (PBA) message
LMA sends a proxy binding acknowledgement (PBA) message
```

Fig 3.4: Multichannel WSN nam file output

3.2 Creation of OSCAR Algorithm

OSCAR [87] uses the rank idea presented in the routing mechanism for Low Power consumption and Lossy Networks (RPL) [139] to compute proximity to the root. In a nutshell, RPL is an IPv6-based LLN that uses a proactive routing protocol. The RPL architecture is a DODAG, or a Destination Oriented Direct Acyclic Graph. The IPv6 edge router, which links nodes normally to the outside world and from which it may receive management orders, is a common analogy for the DODAG root. A packet is sent to the main node from a node by way of a lower-rank neighbor node. An increasing rank is given to each node in the network as it moves away from the centre. Slots are allocated to nodes based on their needs, which is what the rescheduling algorithm is for. The busiest nodes are the ones closest to the root. This method, which is envisioned to be implemented as a component of these node's scheduling, proposes to control the amount of time slots assigned to each node based on the rank of the node. Each network node's rank value is retrieved upon startup. The rank value in RPL is used to label classes.

OSCAR reallocates or releases up a slot if a rank's value shifts. As a result, as the node is further from the root, fewer slots are made available. On the other hand, when a node is brought closer to the root, additional slots are made available. As a result, the class number, will change as that node moves about in the network's topology. Each node awakens at the beginning of each period to either send or receive packets by its class ID. Two pairs of neighboring layers also utilize distinct channels to reduce interference. Nodes away from the root, where rank matters, will utilize less power since fewer slots will be allotted to them. Those near the source will consume somewhat more than the average person, although even this increase is small. In addition to reducing latency, this approach has the added benefit of preventing network congestion at the root node. The complete working of OSCAR is presented in Fig. 3.5.

Let C be the collection of scheduling cells in the network, where each cell has its distinctive identifier. Let N signify the overall count of nodes in the network, where each node has a distinct identifier. Let L be the collection of links between nodes, where each link is a tuple (i, j) indicating a link between nodes i and j.

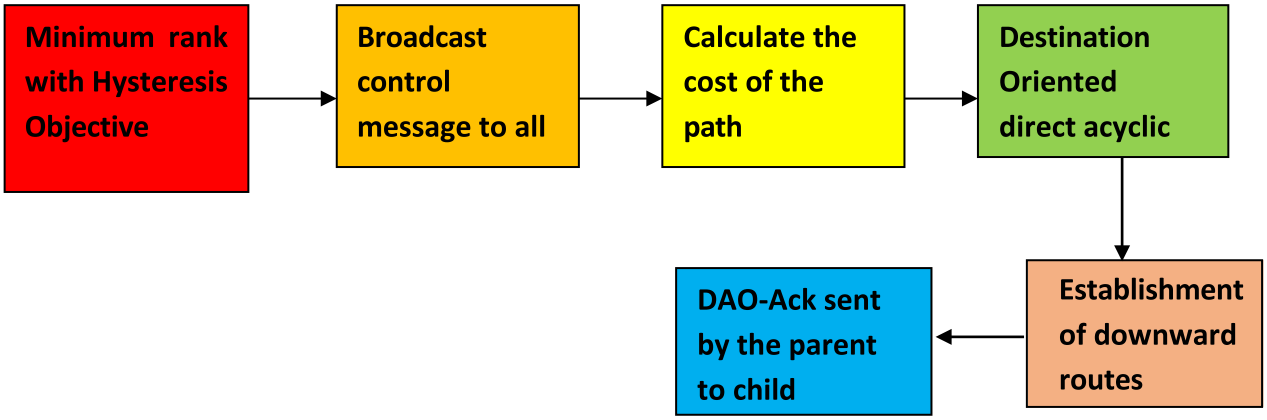


Fig 3.5: Flowchart of OSCAR approach

The goal of the OSCAR Rescheduling Algorithm is to give each node*i* in *N* its scheduling cell *c* in *C* so that the following conditions are met.

- Each cell *c* in *C* is assigned to at most one node *i* in *N*.
- Each node *i* in *N*is assigned exactly one cell *c* in *C*.
- The number of cells in *C* is minimized.

There is a duty cycle that each node in the system follows. Sleep and waking times constitute each period. The node's radio is disabled when it is in the sleep state, and it is activated for packet transmission or reception during the waking state. The orchestra's unicast slot frame has clearly defined regular receiving and broadcast time periods. Throughout their designated receiving time windows, all network nodes display a duty cycle of 100%. In terms of energy output, this is inefficient. This technique maintains a constant slot frame size to meet demand and minimize latency, even under severe traffic conditions. OSCAR's Duty Cycle Adjustment Algorithm uses a mathematical representation to change the duty cycles of each node in the network so that there is less interference, and less energy is used. The mathematical representation of the Duty Cycle Adjustment Algorithm of OSCAR is as follows:

Input: N (set of nodes in the network), *D* (set of duty cycles that can be assigned to nodes).

Output: Di (duty cycle assigned to each node *i*)

- *Step 1:* Initialization:
    - Generate a set of initial duty cycles *D = {d1, d2, ...,dn}*, where each duty cycle is assigned to a node in *N* randomly.
- *Step 2:* Interference Evaluation:
    - Based on the number of collisions between transmissions, figure out how much interference is caused by transmissions from different nodes.
    - Set *C* to be the set of collisions between transmissions in the network, where each collision is represented by a tuple (*i*,j) that shows a collision between nodes *i* and *j*.
- *Step 3:* Duty Cycle Adjustment:
    - Calculate the maximum available time for transmissions *T* in the network.
    - For each node*i* in *N*, figure out the time needed for transmissions *Ti* and the sum(*dj*) of all other nodes' duty cycles (except for node*i*).
    - Calculate the duty cycle assigned to node *i*, *di*, according to the following formula:
- *di* = (*Ti* - *sum(dj*) + *di*) / *T*
- where *Ti* is the time required for transmissions from node *i*, *dj* is the duty cycle assigned to node *j*, and *di* is the current duty cycle assigned to node *i*.
- *Step 4:* Energy Consumption Evaluation:
    - Calculate the energy consumed by each node *i* based on the time required for transmissions *Ti* and the duty cycle assigned to node *i*, *di*.
    - Set *E* to be the set of energy consumed by nodes in the network, where each energy consumption is represented by a tuple (*i, ei*) indicating the energy consumed by node*i*.
- *Step 5:* Termination:

- The algorithm terminates when a solution with the desired level of interference and energy consumption is found a maximum number of iterations is reached.

3.2.1 Limitations of OSCAR

One of the primary challenges associated with OSCAR [87] is its complexity. The algorithm uses a sophisticated optimization technique to allocate scheduling cells for convergecast transmissions. The optimization process involves the use of a genetic algorithm that requires significant computational resources to execute. This complexity can be a major hurdle in the practical implementation of OSCAR, as it can limit the scalability of the algorithm and increase the overhead of the network.

Another challenge associated with OSCAR [87] is its dependence on accurate network information. The algorithm requires precise information about the topology of the network, including the availability of the nodes, the transmission range, and the quality of the wireless link between the nodes. Any errors or inaccuracies in this information can significantly impact the performance of OSCAR, leading to inefficient scheduling and suboptimal use of network resources. Another issue with OSCAR is related to its compatibility with existing protocols and standards. The algorithm was developed specifically for IEEE 802.15.4e TSCH networks, which may limit its applicability to other wireless network environments. Additionally, OSCAR requires modifications to the existing network infrastructure to enable the allocation of scheduling cells, which can be challenging and time-consuming to implement.

One other issue with OSCAR is its vulnerability to network congestion and interference. The algorithm relies on a fixed set of scheduling cells for convergecast transmissions, which can become congested if multiple nodes try to send data at the same time. It may lead to packet loss, increased latency, and decreased network performance. Additionally, interference from other wireless devices can impact the quality of the wireless link, which can further degrade the performance of OSCAR. Finally, the implementation of OSCAR requires significant coordination and collaboration among network nodes. Additionally, OSCAR requires synchronization among all nodes to confirm that convergecast transmissions occur simultaneously, which can be challenging in dynamic network environments.

3.3 Proposed FBESSM Approach

The proposed approach has been presented in this section of the paper. First of all the Cluster–Head (CH) selection process has been defined which is used in the proposed approach which is followed by the data transmission phase as represented in the form of algorithms. The proposed approach is named Fuzzy Based Energy-Efficient Sleep Scheduling Mechanism (FBESSM) where fuzzy logic is integrated for the CH selection based on the sleep scheduling approach for efficient energy utilization. The proposed FBESSM technique has two algorithms in operation as depicted.

The algorithm is a CH (Cluster Head) selection algorithm for WSNs. The goal of the algorithm is to select CHs that will efficiently manage data collection and aggregation tasks in a multi-hop convergecast communication scenario using the fuzzy-based matrix R. The algorithm begins by broadcasting Hello messages to all nodes in the network, establishing communication among them. For each node Ni in the network, the algorithm calculates the distance $D_n(i)$ based on the Received Signal Strength Indicator (RSSI) and adds a Cluster Head Selection (CHS) factor. The CHS factor is used as a threshold for candidate node selection in the later steps of the algorithm. The next step is to elect the CH based on the maximum value of $D_n(i)$ and the residual energy $E_{res}(i)$ of the node.

The algorithm then checks if the nodes can be categorized directly using a fuzzy condition. If not, a Boolean matrix is created using certain rules. The algorithm then randomly selects a candidate node using a value between 0 and 1, denoted as X_0. If X_0 is less than the CHS value of node N_i, the algorithm proceeds with adding the candidate node to the CH. The distance $D_{CH}(i)$ between the CH and the candidate node is calculated, and the information (ID, E_{res}, D_{CH}, D_n) is broadcasted to other nodes, where ID represents the identifier of the candidate node. The algorithm also sets $d_{CH}(i)$ to be the candidate node. On the other hand, if X_0 is greater than or equal to the CHS value of node N_i, the algorithm exits and does not select any candidate node for the current round. Next, the algorithm iterates through the neighbor_SET(S_i) of each node N_i. While receiving CH_msg (Cluster Head message) from node N_i, the algorithm adds the distance d_i to the list of CCHs in *CH_Set*.

If S_i is in the Candidate_CH and the algorithm finds the minimum distance among different CH nodes, it adds node N_i to the *CH_Set*. However, if S_i is not in the Candidate_CH, the algorithm compares the residual energy $E_{res}(j)$ of node N_i with that of the current Cluster Head node $Eres(i)$. If $E_{res}(j)$ is greater than $E_{res}(i)$, Otherwise, the algorithm sends a Control message to the neighbor set using the matrix R. The algorithm continues to iterate through the nodes in the neighbor_SET(S_i) and selects CHs based on the criteria mentioned above until a termination condition is met.

The algorithm above provided is a data transmission algorithm for WSNs in a multi-hop convergecast communication scenario. The algorithm aims to facilitate efficient data transmission from non-CH nodes to CH nodes for further aggregation and processing. The algorithm operates in a time-slotted manner. For each CH_i, the algorithm iterates through each timeslot. In each timeslot, the algorithm checks for control messages received from different sequences of CHs, which contain data and residual energy information. If control messages are received from different CHs with different sequences, the algorithm sends acknowledgement (ACK) messages to all candidate set nodes. This is done to check the reception of data by the CHs.

On the other side, if control messages are not received from different sequences of CHs, the algorithm sends control messages to the CHs containing the *ID*, sequence, and data. This is the actual data transmission process from non-CH nodes to the CHs for aggregation and processing. After completing the transmission to the CHs, the algorithm moves on to non-Cluster nodes. For each non-CH node, the algorithm sends an ACK message containing the *ID* and data. The non-CH nodes then wait for the reception of control_ACK messages from all the non-CH nodes. This is done to ensure that all non- CH nodes have successfully transmitted their data to the CHs and received acknowledgments from other non-CH nodes. The flowchart of working of FBESSM is shown in Fig. 3.6.

Algorithm for Cluster-Head(CH)_selection

```
Broadcast Hello message to all nodes
    $[Y_i]_R=\{Y_j|r_{ij}=1\}$ for individual $Y_i$,the element $y_j$ can be characterized into the same cluster when the fuzzy condition is satisfied
For each node $N_i$
Calculate the distance $D_n(i)$ based on RSSI, add Cluster Head Selection(CHS)
            Elect CH based on $D_{max},E_{res}(i)$
If ($[Y_i]_R$ and $Y_j\,|r_{ij}=1$) are equivalent matrixes the nodes can be categorized directly;
Else
    Matrix $R$ being converted into an equivalent Boolean matrix by certain rules
        Randomly select the candidate node $(0,1)\rightarrow X_0$
If ($X_0$<CHS($N_i$))
                              Add candidate node to Cluster Head
$D_{CH}(i)$=distance between Cluster head to candidate node
                               Broadcast $(ID, E_{res},D_{CH},D_n)$to other nodes; Set
                               $d_{CH}(i)$to the candidate node
Else
Exit(0)
 For each node $N_i$€ neighbor_SET($S_i$)
While receive CH_msg from node $N_i$
                                   Add $di$;
                           If(receives all data samples)
                                             Forward the aggregated result and wake up the dormant nodes
If $S_i$ € Candidate_CH & find minimum distance of different cluster head nodes
                                        Add node $N_i$ to CH_Set;
Else
 If($E_{res}(j)>E_{res}(i)$)
Broadcast highest residual energy node to candidate selection
Else
Send Control message to neighbor set using Matrix $R$
                                  Update state flag and set sleep time interval, then send CHED_MSG_ACK
End If
End if
End if
End While
End for
```

```
Broadcast Hello message to all nodes
$[Y_i]_R = \{Y_j \,|r_{ij} = 1\}$ for individual $Y_i$ ,
the element $y_j$ can be characterized into the same cluster when the fuzzy condition is satisfied
For each node $N_i$
Calculate the distance $D_n(i)$ based on RSSI, add Cluster Head Selection(CHS)
          Elect CH based on $D_{max}$ , $E_{res}(i)$
If($[Y_i]_R$ and $Y_j|r_{ij} = 1$)are equivalent matrixes
the nodes can be categorized directly;
Else
    Matrix $R$ being converted into an equivalent Boolean matrix by certain rules
        Randomly select the candidate node (0,1) → $X_0$
If ($X_0$< CHS ($N_i$)
                         Add candidate node to Cluster Head
$D_{CH}(i)$ = distance between  Cluster head to candidate node
                          Broadcast ( ID, $E_{res}$ , $D_{CH}$, $D_n$ ) to other nodes;
                          Set $d_{CH}(i)$ to the candidate node
Else
Exit(0)
 For each node $N_i$€ neighbor_SET($S_i$)
While receive CH_msg from node $N_i$
                             Add di;
                      If (receives all data samples)
                                     Forward the aggregated result and wakeup the dormant nodes
If $S_i$€ Candidate_CH&& find minimum distance of different cluster head nodes
                                 Add node $N_i$ toCH_Set;
Else
 If ( $E_{res}$ (j)>$E_{res}$(i))
                                             Broadcast highest residual energy node to candidate selection
Else
                            Send Control message to neighbor set using Matrix $R$
                            Update state flag and set sleep time interval, then send CHED_MSG_ACK
End If
 End if
End if
End While
End for
```

```
Algorithm Data_transmission
For each $CH_i$
For each time slot
If (receive control message from different sequence of CH(data,$E_{res}$)
        Send ACK to all candidate set nodes;
 Else
        Send control message to CH(id,sequence,data)
    Endif
End for
End for
For eachnon-cluster
    Send ACK(ID,data)
    Wait
    Receive control_ACK to all non-CH;
End for
```

OSCAR and FBESSM are two separate wireless communication protocols specifically developed to tackle various issues of wireless sensor networks. Although both have the goal of enhancing communication in these networks, their approach and distinctive characteristics differentiate them. OSCAR aims to improve reliability by integrating retransmission methods into its slot and channel assignment procedures. Consequently, if there is any loss or faults in the packets, OSCAR makes an effort to resend the data in order to guarantee its successful transmission. This functionality is essential for applications that prioritize data integrity, particularly in the context of critical infrastructure monitoring. Conversely, FBESSM focuses exclusively on enhancing energy efficiency by utilizing fuzzy logic. FBESSM utilizes fuzzy logic to make judgments by considering imprecise or uncertain data, hence enabling adaptive and intelligent slot allocation for energy conservation. This is particularly advantageous in situations when the power usage of sensor nodes is of utmost importance, such as in extensive sensor networks implemented in distant or remote areas.

3.4 Performance Analysis

We describe the FBESSM's performance in this section. We use measures that we'll go through in this part to compare the performance of FBESSM and OSCAR. We have created and taken into account the scenario while changing the node count to observe how the number of nodes affects performance.

3.4.1 Simulation Results

The proposed framework is implemented using the NS-2 which is a simulator and then the nodes are circulated randomly fashioned in 1000m * 1000m * 1000m area. The counts of nodes are varying from 50 to 200. The count of mobile nodes taken is 10 and the range of transmission of each node considered is 250m. The energy of each node initially is set aside at 100 J. The simulation time is 50 seconds for the results and the packet interval for transmitting the hello packets is kept at 100 seconds. The routing mechanism considered here is AODV [140]. The other simulation factors are shown in Table 3.1.

Choosing the simulation settings for the FBESSM technique is extremely important in order to thoroughly assess its effectiveness in various scenarios. The random topology is used to emulate real-world situations where nodes may be installed without a predetermined pattern, mirroring the dynamic nature of WSN setups. The range of node count, ranging from 50 to 200, is utilized for scalability testing, which aims to assess the protocol's effectiveness over networks of varying sizes. The FBESSM's ability to adapt to different spatial distributions is being tested by setting a node communication range of 250m. Also, most of the research works have used the same simulation settings.

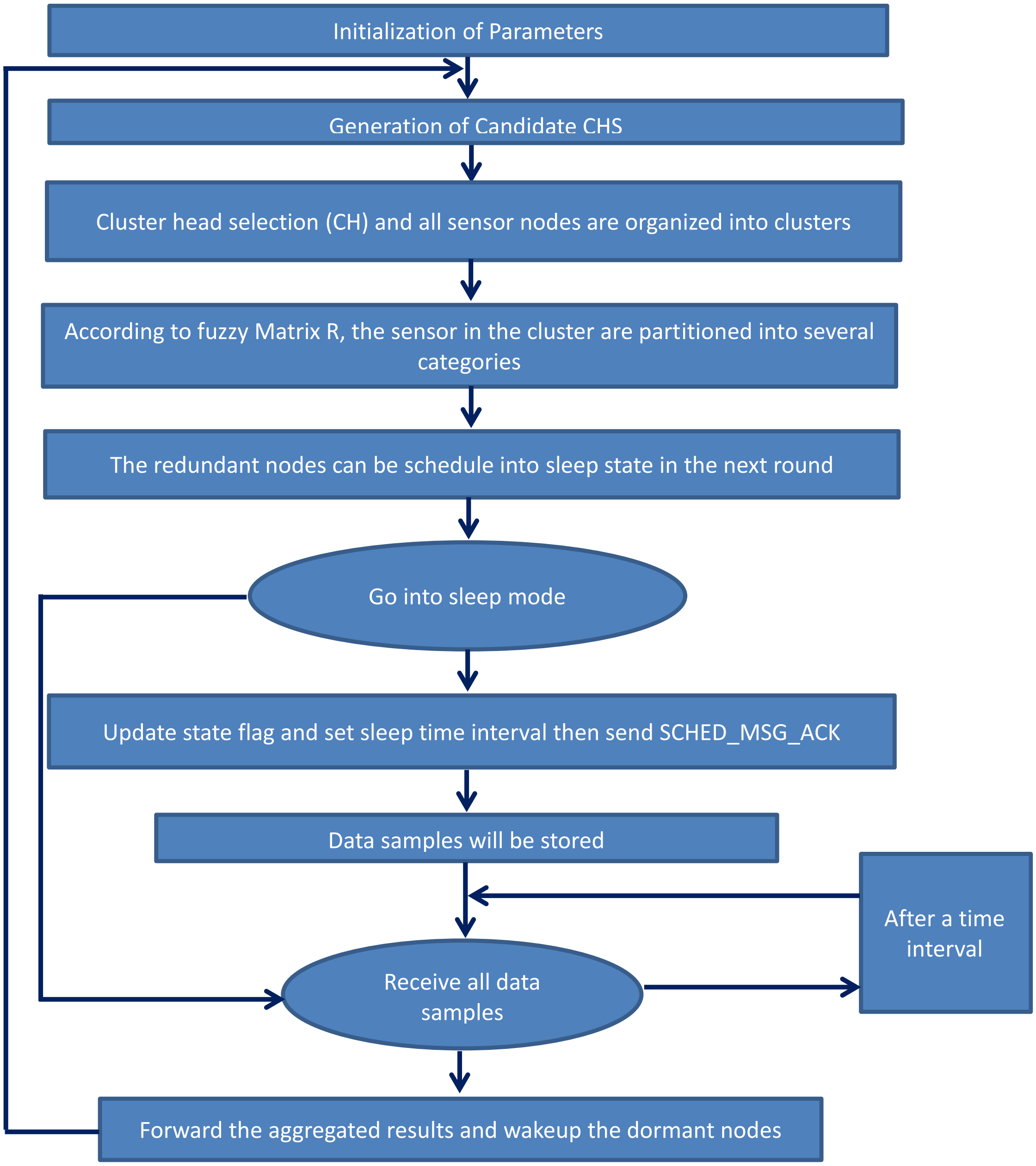


Fig 3.6: Working of FBESSM technique

Table 3.1: Simulation parameters for the proposed FBESSM approach

Simulation Parameter Taken	**Value Taken**
Topology	Random (100m)
Area for Deployment	1000m * 1000m * 1000m
Node count	50- 200
Range	250 m
MAC Protocol	802. 15. 4
Energy	100 J
Radio Propagation Model	Two Ray Ground
Antenna model	Omni Antenna
Number of mobile nodes	10
Routing protocol	AODV
Energy Usage	2w, 0.75w and 10mw
Data Packet Size	64 byte
Time for Simulation	50 s

The MAC protocol is defined as 802.14.4, according to industry standards for low-rate wireless personal area networks (LR-WPANs). The two-ray ground radio propagation model and omnidirectional antenna model accurately represent the radio environment by taking into account ground reflections and the specific properties of Omni antennas. The AODV routing protocol is selected, perhaps due to its widespread use and applicability in ad-hoc and sensor networks. The flexibility of FBESSM to varied energy restrictions, which is an important element in energy-efficient procedures, is assessed by simulating energy use at various power levels (2W, 0.75W, and 10mW). The designated 64-byte data packet size is a prominent choice in several wireless communication contexts. Ultimately, a simulation duration of 50 seconds enables the examination of protocol behavior within an acceptable timeframe, offering valuable insights into both immediate and future long-term performance attributes.

3.4.2 Energy Model

The energy model is required for transmission and reception of data by the nodes intricate in data communication. The quantity of energy spent (E_{trans}) by the transmitting node for transmitting 'T' bits to the receiving node placed at a distance "$Dist$" is given by equations 3.1 and 3.2.

$$E_{trans}(T, Dist) = T * E_{cons} + T * E_{fsm} * Dist^2 \text{ when } Dist < Dist_{thres} \quad (3.1)$$

$$E_{trans}(T, Dist) = T * E_{cons} + T * E_{mam} * Dist^4 \text{ when } Dist_{thres} \leq Dist \quad (3.2)$$

Where E_{cons} indicates the energy consumed while transmitting a single bit, E_{fsm} indicates the energy used under the free space structure or model, E_{mam} indicates the energy utilized under the multipath amplifier structure or model. Also, $Dist_{thres}$ is the threshold value of the distance which is given by equation 3.3.

$$Dist_{thres} = \sqrt{}E_{fsm}/E_{mam} \quad (3.3)$$

Thus based on threshold value ($Dist_{thres}$) the equation 3.1 or 3.2 is selected by the node.

$$E_{receiv} = T * E_{trans} \quad (3.4)$$

3.4.3 Metrics for Evaluation

The factors based on which analysis of the proposed scheme and its comparison with the existing approaches have been done are well-defined below.

Packet Delivery Ratio (PDR) = Σ(Aggregate of packets received by all destination node) /Σ(Aggregate packets send by all source node) (3.5)

End-to-End Delay (EED)= $1/n\Sigma i=1\ (Tr_i - Ts_i) * 1000$ [ms] (3.6)

Where, i= packet identity, Tr_i = time of reception, Ts_i = time of sending, n = successfully delivered packets count

Throughput (TP) = (recvd Size/(stop Time - start Time))*(8/1000) (3.7)

Residual Energy (RE) = Total Energy (TE)- Energy Consumed (EC) (3.8)

Above mentioned factors are very important for the study of the efficiency of the proposed scheme and their comparison with the other existing approaches.

3.4.4 Comparative Evaluation

In this section, the comparative analysis of the proposed FBESSM technique with the existing OSCAR technique has been done based on the metrics as mentioned by varying the node count from 50 to 250. First of all, the OSCAR technique has been implemented on the simulator with the simulation factors as mentioned in Table 1 and then it is tested under the metrics as defined in Table 3.2 while varying the node count. Similarly, in Table 3.3 the proposed technique FBESSM has been tested under the same simulation environment as defined in Table 1 and it's being evaluated with the metrics as defined in Table 3.3.

Table 3.2: Metric-based evaluation for the OSCAR technique

Node Count	Delay	Packet delivery ratio	Throughput	Energy consumption	Network lifetime	Packet loss
50	27.6554	0.8456	338.98	21.4567	45.678	123
100	26.2345	0.8467	341.50	39.5563	34.577	134
150	25.3212	0.8532	347.62	57.3221	26.458	156
200	24.543	0.8598	349.35	76.4567	21.578	178
250	24.542	0.8623	355.98	100.43	17.0872	202

Table 3.3: Metric based evaluation for the proposed FBESSM technique

Node count	**Delay**	**Packet delivery ratio**	**Throughput**	**Energy consumption**	**Network lifetime**	**Packet loss**
50	26.6554	0.8489	352.432	21.321	46.995	120
100	25.2345	0.8543	352.554	38.454	35.665	122
150	24.3212	0.8632	353.443	56.4332	27.404	153
200	24.043	0.8690	355.667	75.433	21.5092	173
250	23.781	0.8721	358.443	99.445	18.455	197

In Fig 3.7 the variation of EED for the node count 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. It has been depicted that FBESSM technique outperforms the OSCAR technique for EED at all instances. Delay is reduced with OSCAR, and this is explained by the reason that the congested nodes are given extra time slots to clear out their congestion. But the EED in FBESSM is less on account of the fuzzy matrix used for the efficient CH selection based on sleep scheduling approach as less overhead while communication. FBESSM reduces EED by 0.63%, 2.78%, 3.94%, 2.03% and 3.10% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

In Fig 3.8 the variation of EC for the node count 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. It has been depicted that FBESSM technique is performing slightly better than the OSCAR technique. Since certain nodes are less engaged than others, OSCAR has a method for reducing the number of slots allotted to inactive nodes, which results in less EC overall. On the other hand FBESSM is using the concept of sleep scheduling of nodes which are not participating in the CH selection process and thus reduces the EC to large extent. FBESSM reduces EC by 3.61%, 3.81%, 1.55%, 1.33% and 0.98% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

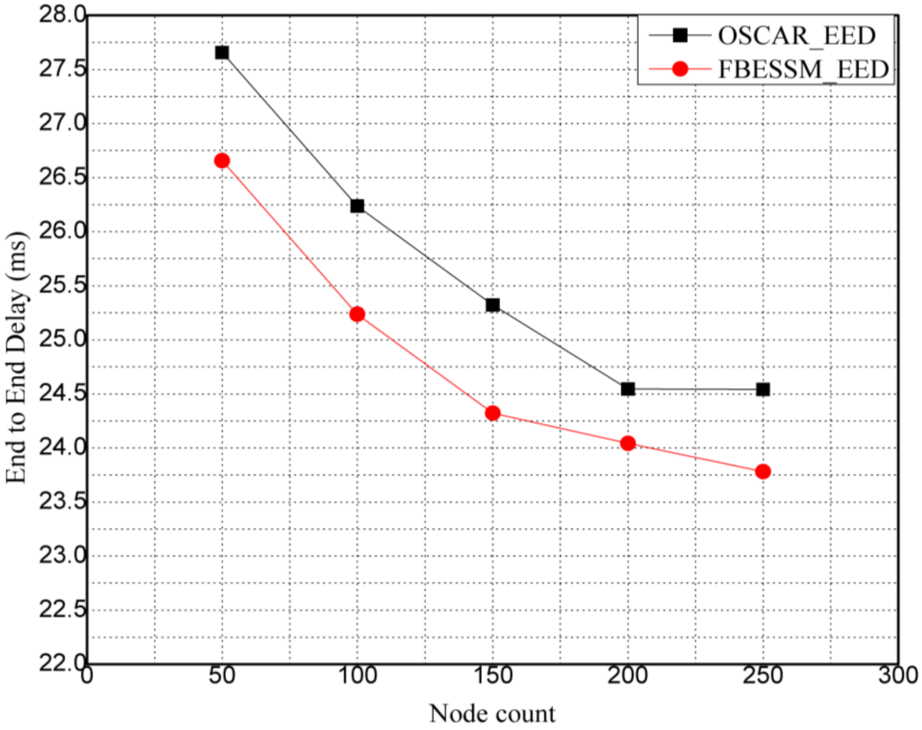


Fig 3.7: Variation of EED under the node count

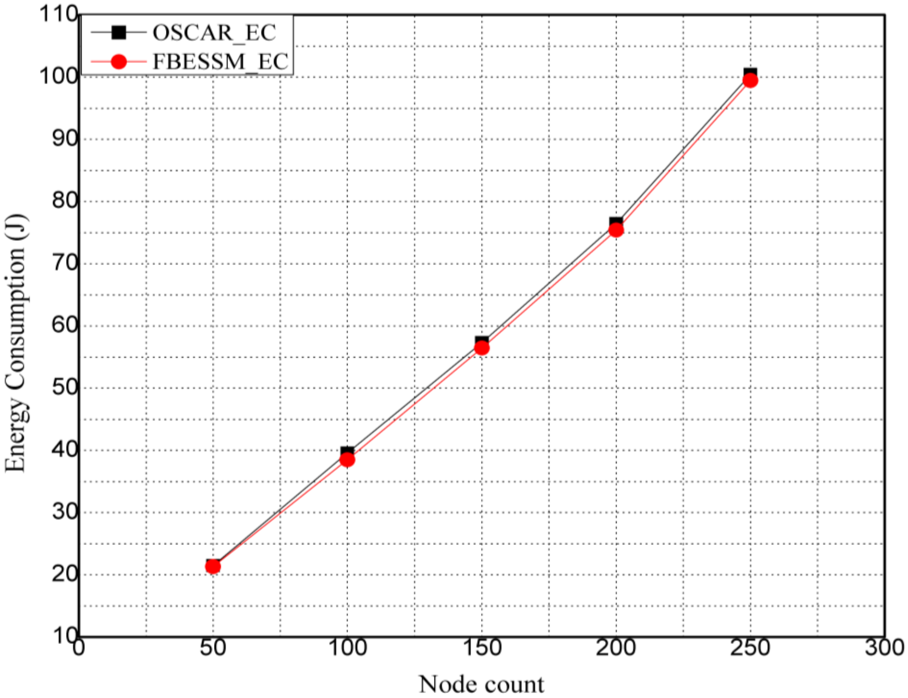


Fig 3.8: Variation of EC under the node count

In Fig 3.9 the variation of Network Life Time (NLT) for the node count 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. Because it takes into account several factors including RE and distance to BS when choosing a CH node, FBESSM consistently outperforms competing algorithms across all test situations. FBESSM increases NLT by 2.88%, 3.14%, 3.57%, 0.31% and 0.80% in comparison to OSCAR under node count 50, 100, 150, 200, and 250 respectively.

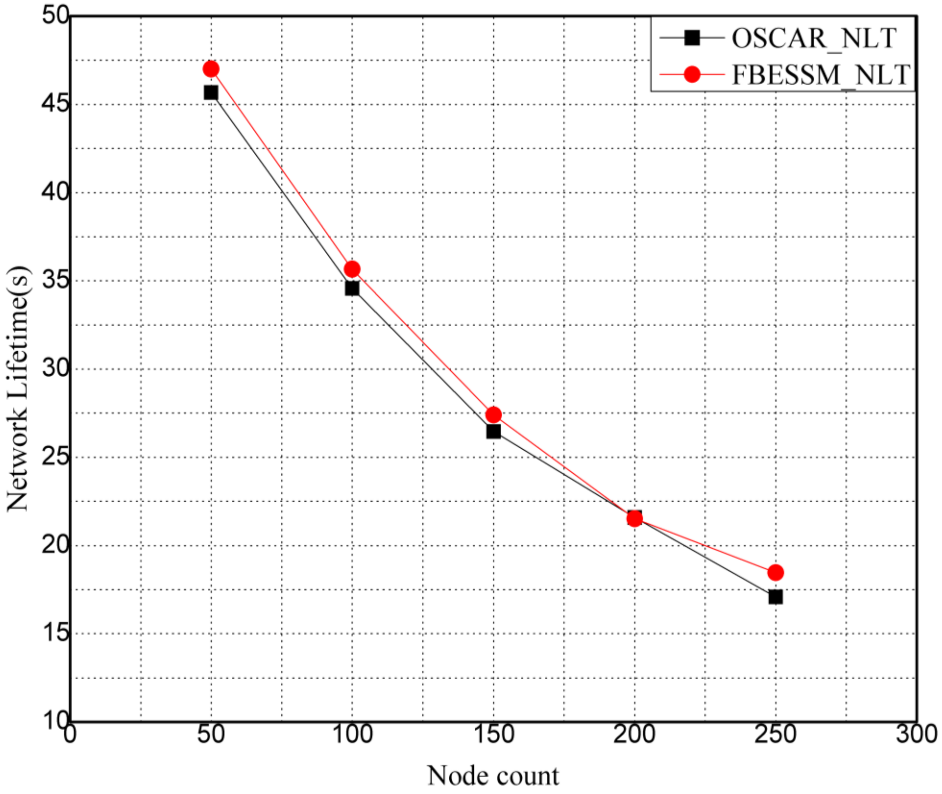


Fig 3.9: Variation of NLT under the node count

From Fig 3.10, it is depicted that the PDR of FBESSM is quite high in contrast to OSCAR technique when the node count is increased from 50 to 250., FBESSM may achieve higher PDRs by reducing the probability of packet collisions and interference. FBESSM has higher PDR of 0.39%, 0.89%, 1.17%, 1.07% and 1.13% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively. In a similar fashion, the packet loss variation has been shown in Fig 3.11 for the FBESSM in contrast to OSCAR. FBESSM has lower packet loss by 2.4%, 8.9%, 1.92%, 2.8% and 2.4% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

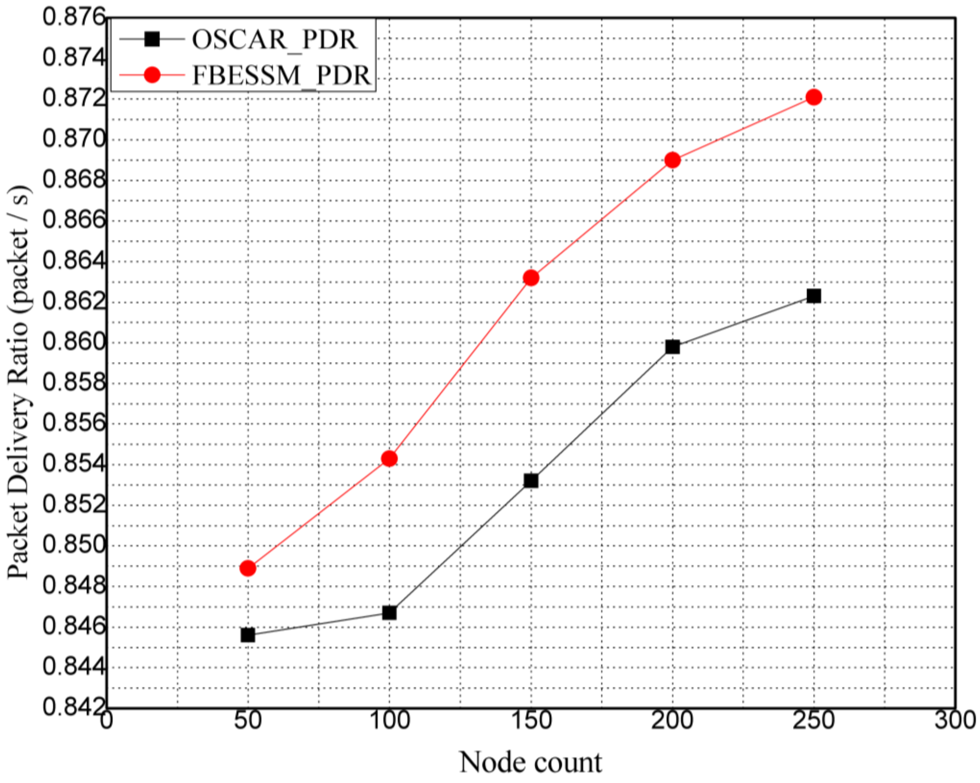


Fig 3.10: Variation of PDR under the node count

Also, in Fig 3.12 throughput of OSCAR is compared with FBESSM by varying the node count from 50 to 250. It is depicted that FBESSM has higher throughput in contrast to OSCAR technique. FBESSM has higher throughput by 3.9%, 3.2%, 1.6%, 1.8% and 0.69% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

FBESSM majorly caters to real-time situations, especially in WSNs. The device's effective slot management system makes it ideal for tasks such as environmental monitoring, where sensors are spread across extensive areas and must function with limited energy resources. Moreover, FBESSM has the capability to improve the effectiveness of intelligent agricultural systems by allowing uninterrupted surveillance of soil conditions and crop health.

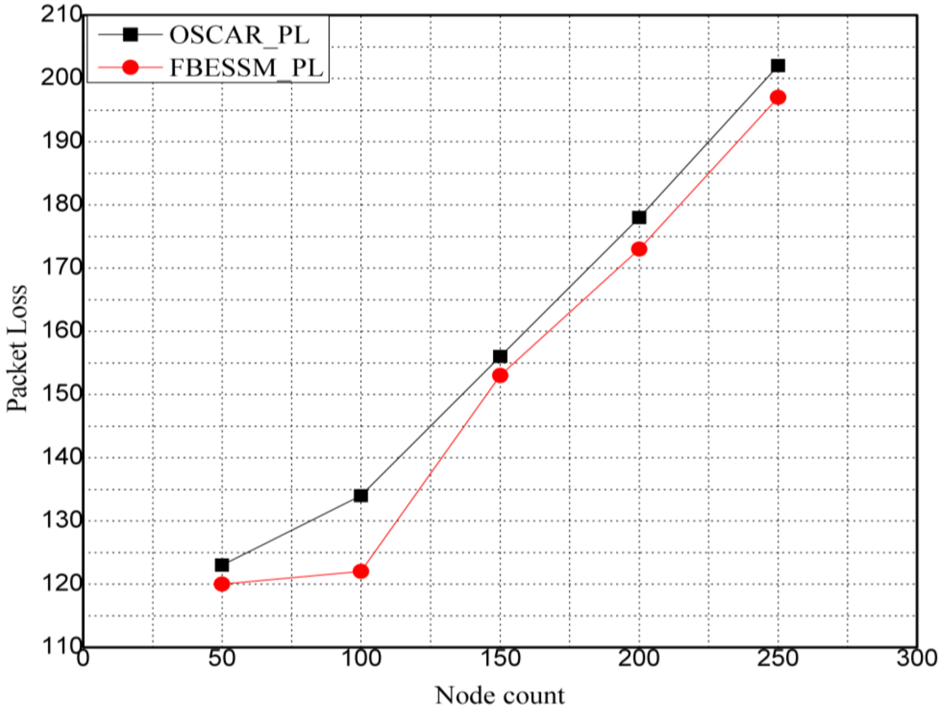


Fig 3.11: Variation of packet loss under the node count

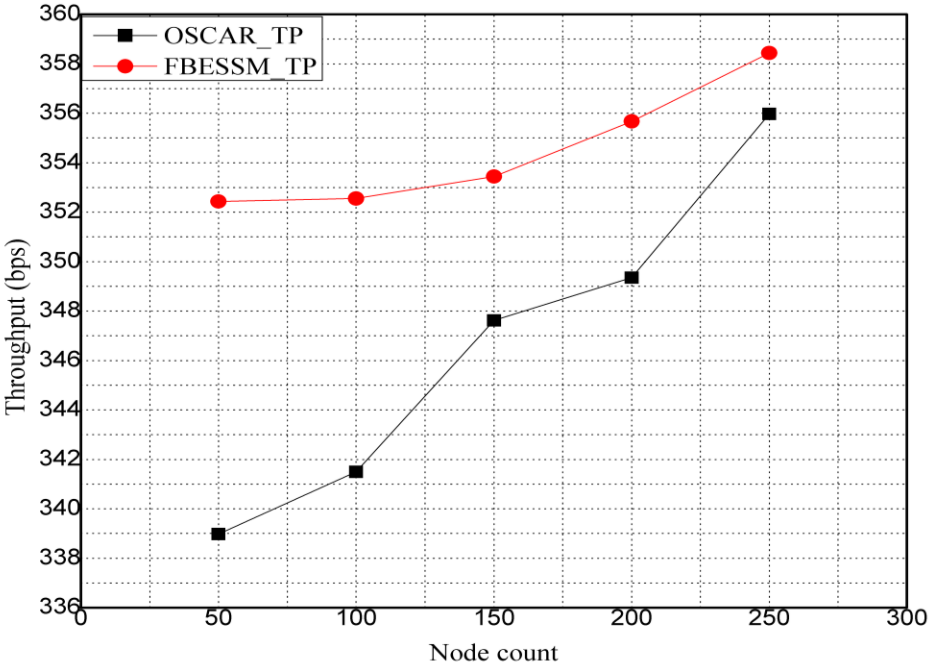


Fig 3.12: Variation of throughput under the node count

In extremely large-scale networks, FBESSM may face performance challenges due to increased complexity in managing numerous nodes and making real-time fuzzy logic decisions. The scalability of the algorithm could be hindered by the need for extensive coordination and communication overhead. However, its energy-efficient approach could still provide substantial benefits, reducing overall power consumption and extending network lifetime, provided the fuzzy logic rules are optimized for large-scale deployments.

Simulation-based evaluation of FBESSM faces challenges in real-world validation due to the inability to fully replicate environmental dynamics, hardware constraints, and unpredictable network behaviors. Simulations often overlook factors like interference, physical obstacles, and node failures, leading to discrepancies in performance outcomes. Additionally, real-world energy consumption and latency might differ, complicating the accurate assessment of FBESSM's practical efficacy.

FBESSM, in the framework of the Internet of Things (IoT) in industrial settings, may enhance communication between sensors. This optimization results in dependable and energy-efficient transfer of data, which is beneficial for activities like predictive maintenance and process optimization. FBESSM can be utilized in the field of home automation to enhance communication among intelligent devices, including thermostats, lighting controls, and security sensors. The energy-efficient slot management method guarantees reliable communication and saving energy, so extending device battery life and promoting overall system efficiency. FBESSM is highly beneficial for vehicle communication networks inside transportation systems. It can improve the effectiveness of data communication between automobiles, traffic signals, and road infrastructure, facilitating real-time traffic control and accident avoidance. FBESSM enhances transportation safety and reduces congestion by dynamically adapting communication slots according to traffic conditions.

FBESSM can be utilized in precision agriculture to facilitate effective communication among diverse agricultural sensors, aerial vehicles, and automated machines. This enhances the gathering and examination of up-to-the-minute data on the state of the soil, crop health, and weather patterns, resulting in improved decisions for crop irrigation, fertilization, and pest management. The energy-efficient slot management system developed by FBESSM is especially advantageous in this scenario, as it allows for the effective deployment of devices in remote locations with restricted access to power sources. FBESSM facilitates communication among smart meters, grid devices, and control centers in the domain of smart grids. The fuzzy-based energy-efficient strategy guarantees both strong communication and little energy usage, enabling real-time monitoring and management of the electrical grid. Enhancing the efficiency and dependability of electricity distribution networks in smart grid installations is important.

FBESSM faces potential limitations such as the complexity of implementing and tuning fuzzy logic rules for optimal performance in diverse network conditions. It may struggle with scalability in large networks due to increased coordination overhead. Additionally, real-time adaptability to dynamic network changes can be challenging, potentially impacting energy efficiency and reliability. The computational requirements for fuzzy logic processing may also strain resource-constrained nodes.

3.5 Summary

This chapter proposed a fuzzy based cluster selection approach which is energy efficient also through sleep mode mechanism for convergecast WSN. The technique has two major prominent phases. The first phase is the utilization of a fuzzy matrix to select a Cluster Head (CH) is rely on distance and residual energy. The second phase is the data transmission method which involves the control messages. This phase facilitates data gathering and aggregation by transmitting data efficiently in a multi-hop convergent communication environment. When comparing FBESSM with OSACR on EED, performance analysis highlighted that FBESSM is superior. Configurations of the FBESSM and OSCAR under different metrics and with node counts ranging from 50 to 250 have been explored to get insight into their performance. While both FBESSM and OSCAR assign slots, the experimental findings reveal that FBESSM's allocation of slots based on a fuzzy matrix leads to greater overall performance. In addition, simulations with varying numbers of nodes have shown FBESSM's superior performance on large networks. Increases in network capacity and associated traffic load lead to improved network lifetime and packet delivery under FBESSM compared to OSCAR. It is also observed that the FBESSM significantly decreases the EC by using the notion of sleep scheduling for nodes which are not participated in the CH selection process. When comparing EC usage with OSCAR, FBESSM achieves 3.61 %, 3.81 %, 1.55 %, 1.33 %, and 0.98% lower EC against the node counts of 50, 100, 150, 200, and 250, respectively. Similar kind of results has been inferred for the other metric evaluation like packet loss, throughput etc.

CHAPTER 4

EMCSABA: EFFICIENT MULTI CHANNEL AND SLOT ALLOCATION WITH BELIEF SCORE AUTHENTICATION

Channel allocation is one of the major concerns in WSNs. Since WSNs frequently operate in a spectrum-constrained context, poor channel selection can result in signal degradation, collisions, energy waste, and the precision of data delivery. By leveraging multiple channels, simultaneous transmissions can be performed, enhancing network capacity and reducing contention. Additionally, optimized slot allocation ensures efficient resource utilization and minimizes delays. The major concerns in applying the approach of efficient multi- channel and slot allocation include the complexity of channel allocation algorithms, synchronization issues among nodes, and overhead associated. To address these issues, in this chapter a new approach is presented which is a belief score-based approach for effective slot allocation in a multichannel environment. Incorporating a belief score authentication scheme strengthens the security of data aggregation, allowing only trusted nodes to participate in the convergecast process. Performance parameters such as throughput, packet loss, lifespan, energy consumption, and latency are being used to assess the effectiveness of the Efficient Multi Channel and Slot Allocation with Belief Score Authentication (EMCSABA) approach as it has been implemented in NS-2.

4.1 Slot Allocation in Multichannel WSN

Slot allocation in multichannel WSNs can be done using various algorithms and techniques. One common approach is Time Division Multiple Access (TDMA), where time is divided into slots and every node is allocated for transmission. In multichannel scenarios, multiple channels are used to increase network capacity. Slot allocation algorithms consider factors such as channel availability, node priorities, traffic patterns, and interference levels to efficiently assign slots and channels to nodes. Dynamic slot allocation algorithms, such as distributed scheduling algorithms or centralized controllers, may be employed to adaptively allocate slots based on changing network conditions and communication requirements.

In this context, there are two common slot allocation approaches: centralized and distributed, along with examples to illustrate their operation.

- **Centralized Slot Allocation:** In a centralized slot allocation scheme, a central controller or sink node is responsible for assigning slots to the participating nodes in the network or system. The central controller maintains information about node priorities, channel availability, and traffic patterns to make efficient slot allocation decisions.
- **Distributed Slot Allocation:** In a distributed slot allocation scheme, the slot allocation decision is made by the participating nodes themselves, without the involvement of a central controller. Nodes exchange control messages and collaborate to allocate slots based on predefined rules or algorithms. The distributed slot allocation scheme offers decentralized decision-making, reducing the dependency on a central controller and improving network resilience. However, it requires efficient coordination and communication among nodes, as well as robust slot allocation protocols to avoid conflicts and achieve optimal slot utilization. Both centralized and distributed slot allocation approaches have their advantages and considerations. The choice between them depends on factors such as network size, resource constraints, scalability, and the desired level of control and coordination in the network.

4.2 Proposed Scheme for Efficient Slot Allocation

The proposed EMCSABA approach involves selecting nodes based on their number of neighbors, assigning channels to minimize interference, and allocating time slots based on proximity to the sink. The algorithm utilizes a centralized or distributed approach for slot allocation. The goal is to ensure efficient data collection, minimize collisions, and optimize network throughput. It has two parts: a) Efficient multichannel and slot Allocation and b) Efficient slot acquisition.

4.2.1 Efficient Multichannel and Slot Allocation

Here the proposed algorithm describes an efficient multichannel and slot allocation approach for a convergecast scenario in a WSN. Here's a step-by-step explanation of the proposed algorithm:

1. *Input:*
    - X: Graph representing the WSN (vertices and edges)
    - Gc: Graph representing the communication channels (vertices and edges)
    - L: Length of time slot in a period
    - Sch: Set of available channels
2. *Initialization:*
    - Set all nodes' channels in Vc (vertices in the communication graph) as unselected
    - Set all nodes' time slots in Vc as infinity (∞)
3. While there are nodes U in Vc with unselected channels:
    - Identify the set of nodes M in Vc such that for each node V in M, the number of neighbors of V in X is less than or equal to L. (M is the set of nodes eligible for selection)
    - Choose any node U from M such that the number of neighbors of U in X is maximum among the nodes in M.
    - Determine the set D of channels for U to avoid interference. D consists of the channels assigned to U's neighbors in Gc.
    - Assign the minimum channel from Sch/D (division of available channels by D) to U.
    - For each neighbor V of U in X:
    - If V's channel is unselected, assign it the same channel as U.
4. Set the current time (Curr_time) to 1.
5. Bottom-up approach:
    - While there exist nodes V in U's neighbors such that V's timeslot is infinity (∞):

- Choose any node V from U’s neighbors where V’s timeslot is NULL.
- If V has no child nodes in X (indicating it is closer to the sink):
- Assign V’s timeslot as Curr_time and increment Curr_time by 1.
- Else, if there exists a node W in V’s neighbors such that W’s timeslot is unselected:
- Assign W’s timeslot as Curr_time and increment Curr_time by 1.
- Remove U from U’s neighbors in X.
- Update the neighbor relationship of V from U to SINK in X.

This algorithm aims to allocate different channels and time slots to nodes for minimizes interference, ensures efficient utilization of channels, and enables reliable convergecast in the WSN. It employs a bottom-up approach to assign time slots based on proximity to the sink, enabling data collection in an organized manner. The pseudo-code representation of this approach has been depicted on the next page.

The algorithm begins by taking inputs, including the graph X representing the wireless sensor network (WSN) with its vertices (V) and edges (E_T), the communication channel graph Gc with vertices (V_c) and edges (E_c), the length of time slot (L) in a period, and the set of available channels (S_{ch}). Next, it initializes all nodes in V_c with unselected channels (U.channel) and infinite time slots (U.time). The algorithm enters a loop that continues until there are nodes in V_c with Within the loop, it identifies a set of nodes (M) in V_c whose number of neighbors in X is less than or equal to L. It then chooses a node (U) from M with the maximum number of neighbors. The algorithm calculates a set of channels (D) that are not assigned to U’s neighbors in G_c to avoid interference. U’s channel is set to the minimum channel value obtained by dividing S_{ch} by D.

For each neighbor (V) of U in X, if *V’s* channel is unselected, it is assigned the same channel as U. The current time (Curr_time) is set to 1. The algorithm continues with a bottom-up approach, looping until there are nodes (V) in U’s neighbors whose time slot is still infinity (∞). It selects any node (V) from U’s neighbors with a NULL time slot. If V has no child nodes in X, it assigns V’s time slot as Curr_time and increments Curr_time by 1.

```
Input: X(V,$E_T$) $G_c$ =($V_c$, $E_c$),L,S$c_h$

V- Vertex    $E_T$ -> Edge L-length of time slot in period  $S_{ch}$→ Channel set

¥ U ∑ V   U.channel<- unselected ,u.time← ∞

While{ U € $V_c$ | u.channel  == unselected } ≠ 0

        M ←{ V€ $V_c$ || V.neighbor (X) | <= L}

                  //M –Set of nodes

        U← choose any node such that  |a.neighbor(X) == max{ V.neighbor(X) || V€ M}

        D ← {V.channel | V€ u.Distance ($G_c$)}

        //D  ← Set of channels for avoid interference

u.channel← min( $S_{ch}$/D) // lowest channel

for each V€u.neighbor(X)

if V.channel == unselected

V.channel←u.channel // allocate same channel to neighbor nodes

Curr_time← 1

//Bottom up approach

While { V€ u,neighbor(X) | V.timeslot == ∞} ≠ 0

      V ← Choose any node in {V € u.neighbor(X) | V.timeslot == NULL}

If (V.child(X)==0)

v.timeslot←cur_time ++

else if (W € V.neighbor (X) | W.timeslot == unselected } ≠0

w.timeslot<- curr_time ++

removeu.neighbor(X)
```

Otherwise, if there exists a node (*W*) in V's neighbors with an unselected time slot, *W*'s time slot is set as Curr_time and incremented by 1. After that, the algorithm removes *U*'s neighbors from *X* and updates the neighbor relationship of *V* from *U* to the SINK node in *X*. The above algorithm repeats this process until all nodes have been assigned channels and time slots, ensuring efficient multi-channel and slot allocation for convergecast in the WSN.

4.2.2 Efficient Slot Acquisition

The efficient slot acquisition algorithm aims to allocate a time slot to a new node (*n*) in a wireless network. The algorithm takes inputs such as the new node (*n*), its one-hop neighbor set (N_n), and the set of available time slots (*Tn*). The algorithm's output is the reservation of a time slot (M_x) for the node *n*. The algorithm starts by checking if the set of available time slots (T_n) is not empty. If it is not empty, the algorithm proceeds with the following steps. The algorithm ensures efficient slot acquisition by attempting to allocate a time slot without conflicts. If conflicts arise, the algorithm utilizes the CSMA protocol for collision avoidance and node discovery, allowing node n to find a suitable and unoccupied time slot.

The pseudo-code representation of this approach is as follows. The proposed slot acquisition scheme operates as follows: Once a node (referred to as node x) receives all the packets from its one-hop neighbors, it proceeds to check if there are any available time slots for occupancy. If there are indeed available time slots, node x randomly selects one and decides to occupy it. In the subsequent TDMA period, node x broadcasts its packet in the chosen time slot. For successful occupancy, it requires all one-hop neighbors of node x to add its identification and modify the related bit in the bitmap within their neighbor information related to the received packets.

This ensures that node *x*'s occupancy request is acknowledged by all neighbors. However, if the neighbors do not comply with the occupancy request, node *x* selects another available time slot and repeats the process. If all available time slots have been exhausted, node *x* proceeds to explore the next option.

Efficient Slot acquisition algorithm

```
Input n: New node
Nn: one hop neighbor set
Tn: Set of available time slot of node n
MANC: Packet transmitted during TDMA time slot
Output Mx: Reservation of timeslot of node n While Tn
is not empty
tn←Tn;
if n and tn (MANCy,¥y€ Nn)
                  Node n successfully occupies tx
break;else
Tx=Tx/tx;
END
if Tx is empty
    Node n broadcasts HELLO packet CSMA protocol
While y€Nn, x and tx€MANy
             Node x broadcasts a Hello packet N CSMA period
      END
```

- Select a time slot (t_n) from T_n.
- Check if there is no conflict between node n and the existing nodes in terms of time slot occupancy. If there is no conflict (i.e., nodes *n* and t_n do not overlap with any existing node's occupied time slots in MANC), node n successfully occupies time slot t_n, and the algorithm breaks.
- If there is a conflict, update T_n by removing t_n from the set of available timeslots.
- Repeat steps 1to3 until all time slots in T_n are checked.
- If T_n becomes empty, indicating that all available time slots have been checked and there is a conflict with the existing nodes, node n initiates a broadcast of a HELLO packet using the CSMA protocol. This HELLO packet informs other nodes about node n's presence.
- While node '*n*'sone-hop neighbors(N_n) exist, and there are other nodes(y)in N_n, and their time slots (t_x) and MANC periods (MANy) exist

In the scenario where no available time slots exist, node *x* resorts to broadcasting a HELLO packet during the CSMA period. In order for node *x* to successfully occupy an extended time slot, it requires all one-hop neighbors to add its identification and modify their neighbor information to indicate the extended time slot. If node *x* fails to initially occupy a time slot, it persists in broadcasting HELLO packets until it ultimately succeeds in acquiring a time slot.

EMCSABA faces algorithmic obstacles in effectively handling the dynamic allocation of channels and slots, particularly when dealing with changing network demands. Synchronization problems can occur when there is a requirement for accurate timing synchronization among multiple nodes, resulting in possible delays and collisions. Implementing real-time belief score authentication introduces complexity, which could potentially affect the overall performance and security of the network in practical scenarios.

4.3 Performance analysis of EMCSABA algorithm

This section describes the EMCSABA’s performance. We use measures that we’ll go through in this part to compare the performance of EMCSABA with FBESSM (proposed) and OSCAR [87]. We have created and taken into account two scenarios, one while changing the node count to observe how the number of nodes affects performance and the other one based on varying simulation time. EMCSABA relies heavily on scalability and resilience, as these factors determine the protocol's capacity to manage extensive and ever-changing networks while sustaining optimal performance even in challenging circumstances. If these elements are not thoroughly evaluated, there is a possibility of encountering issues such as higher latency, greater coordination overhead, and susceptibility to node failures. These concerns can have a negative impact on the overall efficiency and dependability of the network. But the same will be done in the future aspects.

EMCSABA's performance is greatly influenced by system characteristics such as node density, communication range, and channel availability. Increased node density amplifies the difficulty of coordination and the likelihood of collisions, while a restricted communication range impacts the interdependence of the network. The effectiveness of slot allocation and overall throughput is affected by the availability of channels. An in-depth examination of these characteristics is essential for maximizing EMCSABA under diverse network situations.

4.3.1 Comparative analysis based on node count

In this section, the comparative analysis of the proposed EMCSABA technique with the existing FBESSM and OSCAR techniques has been done based on the metrics mentioned by varying the node count from 25 to 400. First of all, the EMCSABA technique has been implemented on the simulator with the simulation factors as mentioned in Table 3.1 of Chapter 3.

In Fig. 4.1 the variation of EC when the node count is varied from 25 to 100 has been presented and the comparative analysis of EMCSABA has been done with FBESSM and OSCAR. It has been depicted that the EMCSABA technique is performing quite better than the FBESSM and OSCAR techniques. Since certain nodes are less engaged than others, OSCAR has a method for reducing the number of slots allotted to inactive nodes, which results in less EC. On the other hand, FBESSM uses the concept of sleep scheduling of nodes which are not participating in the CH selection process and thus reduces the EC to a large extent as compared to OSCAR. Whereas in EMCSABA node authentication using the belief score mechanism leads to less consumption of energy owing to non-participation of unauthenticated nodes and thus a more effective one. EMCSABA reduces EC by 48 %, 55 %, 61 %, and 71 % in comparison to OSCAR under node count 25, 50, 75, and 100 respectively. Also, EMCSABA reduces EC by 34 %, 45 %, 54 % and 11 % in comparison to FBESSM under node counts 25, 50, 75, and 100 respectively.

In Fig. 4.2 the variation of Network Life Time (NLT) when the node count is varied from 50 to 400 has been presented and the comparative analysis of EMCSABA has been done with FBESSM and OSCAR. By intelligently managing energy usage, EMCSABA extends the overall network lifetime. EMCSABA increases NLT by 20.2 %, 17.1 %, 17.3%, and 20.5 % in comparison to OSCAR under node count 25, 50, 75, and 100 respectively. Also, EMCSABA increases NLT by 11.7 %, 19.7 %, 20.5 %, and 12.8 % in comparison to FBESSM under node count 25, 50, 75, and 100 respectively.

From Fig. 4.3 it is depicted that the PDR of EMCSABA is quite high in contrast to OSCAR technique and slightly better in comparison to FBESSM when the node count is increased from 25 to 100. The reason is that the FBESSM incorporates fuzzy logic-based decision-making mechanisms to optimize energy consumption by controlling the sleep/wake cycle of the nodes. This allows the nodes to stay in sleep mode for longer periods, which reduces the amount of communication overhead and collisions in the network. EMCSABA has a higher PDR of 34.7 %, 19.5 %, 16.6 %, and 11.23% in comparison to OSCAR under node count 25, 50, 75, and 100 respectively. While in comparison to FBESSM, PDR increments of 1.56 %, 1.87%, 13.96%, and 2.04% under node count 25, 50, 75 and 100 respectively.

From Fig. 4.4 it is depicted that the RE of EMCSABA is high in contrast to OSCAR technique and slightly better in comparison to FBESSM when the node count is increased from 25 to 100. EMCSABA exhibits higher residual energy compared to FBESSM and

OSCAR when varying the node count due to its efficient energy utilization and distribution mechanisms. EMCSABA has higher RE of 6.38 %, 12.38 %, 8.11 %, and 2.70% in comparison to OSCAR under node count 25, 50, 75, and 100 respectively. While in comparison to FBESSM, RE increments of 2.04 %, 2.61%, 3.90%, and 1.06% under node count 25, 50, 75, and 100 respectively.

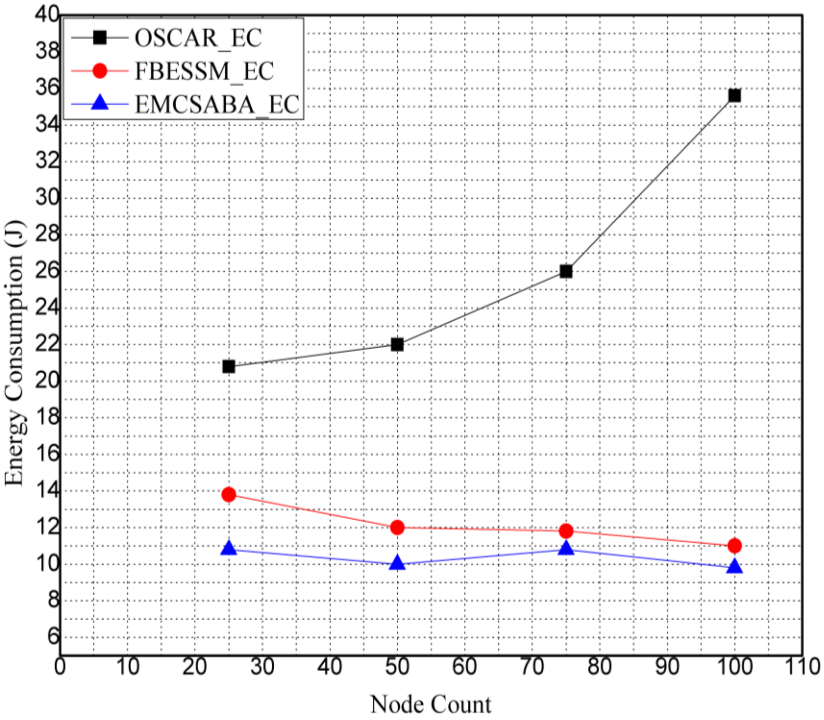


Fig 4.1: Variation of EC under the node count

4.3.2 Comparative analysis based on simulation time

In this section, the comparative analysis of the proposed EMCSABA technique with the existing FBESSM and OSCAR techniques has been done based on the metrics as mentioned by varying the simulation time from 50 to 200 seconds. First of all the EMCSABA technique has been implemented on the simulator with the simulation factors as mentioned in Table 3.1.

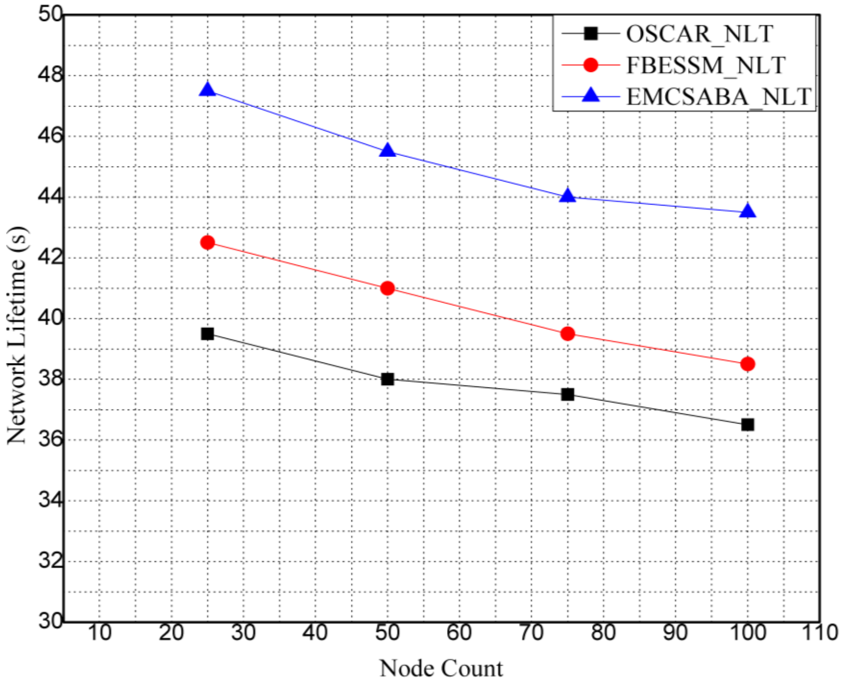


Fig 4.2: Variation of NLT under the node count

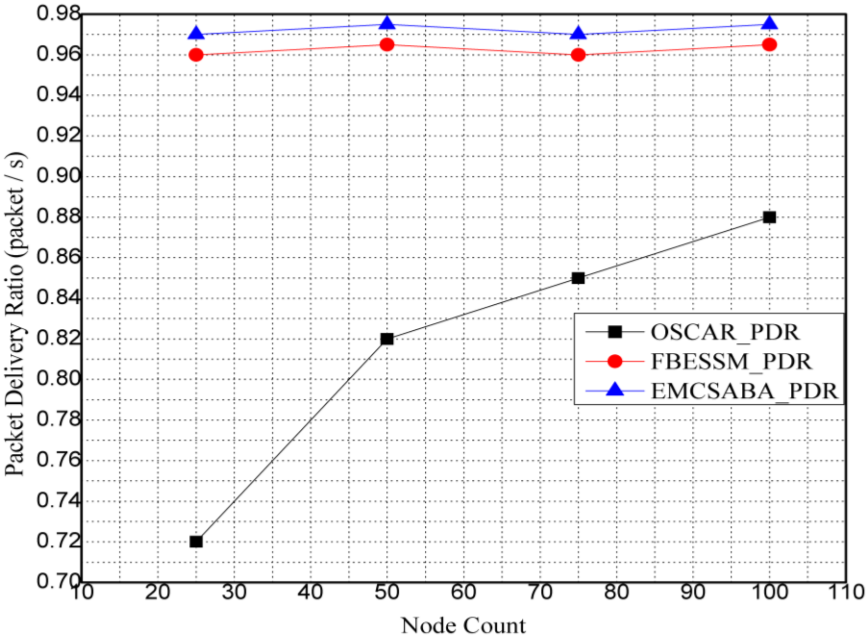


Fig 4.3: Variation of PDR under the node count

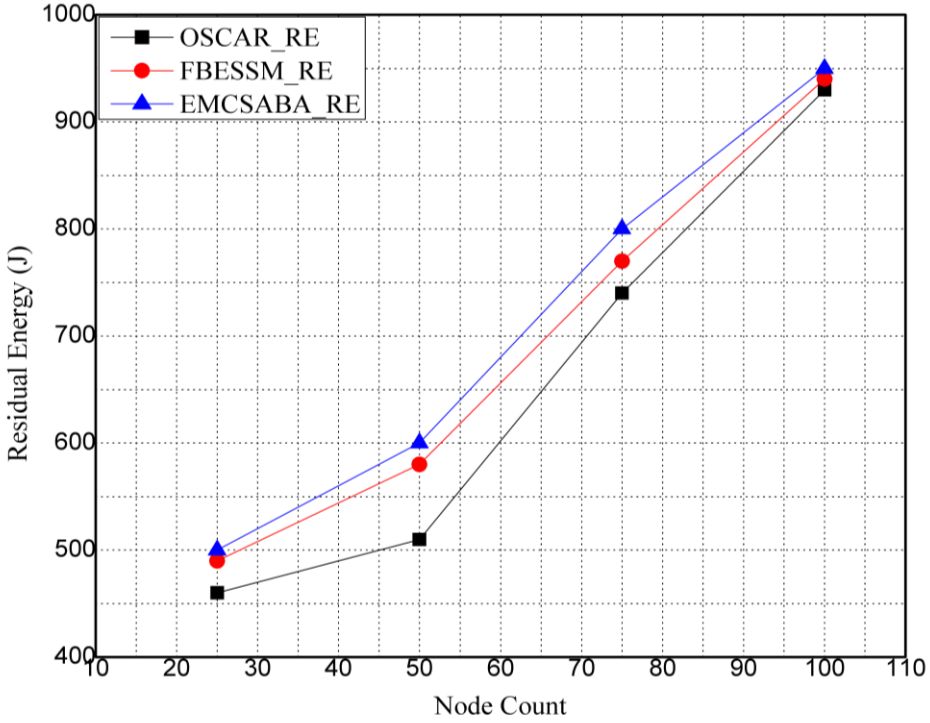


Fig 4.4: Variation of RE under the node count

In Fig. 4.5 the variation of EED when the simulation time is varied from 50 to 200 seconds has been presented and the comparative analysis of EMCSABA has been done with FBESSM and OSCAR. It has been depicted that FBESSM technique outperforms the OSCAR technique for EED at all instances. Delay is reduced with OSCAR, and this is explained by the reason that the congested nodes are given extra time slots to clear out their congestion.

By intelligently managing network resources and minimizing idle times, EMCSABA ensures faster and timelier data delivery, resulting in a lower EED compared to FBESSM and OSCAR. EMCSABA reduces EED by 3.92%, 4.12%, 4.27% and 6.64 % in comparison to FBESSM under simulation time 50, 100, 150 and 200 seconds respectively. In Fig. 4.6 the variation of EED when the simulation time is varied from 50 to 200 seconds has been presented and the comparative analysis of EMCSABA has been done with FBESSM and OSCAR. EMCSABA optimizes energy usage by efficiently allocating channels and time slots, minimizing unnecessary energy expenditure.

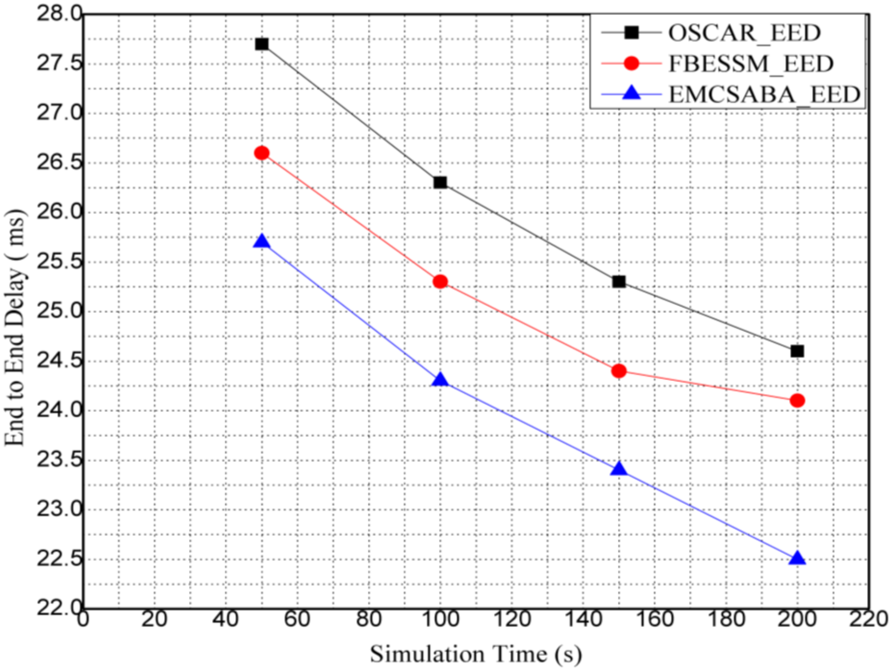


Fig 4.5: Variation of delay under the simulation time

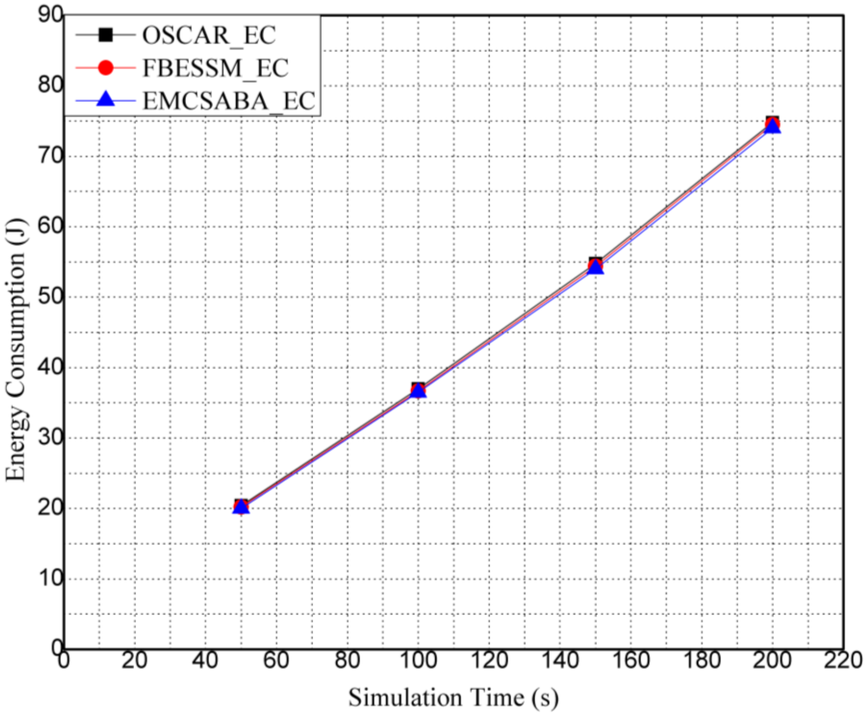


Fig 4.6: Variation of EC under the simulation time

Additionally, its belief score authentication mechanism reduces energy consumption by eliminating unnecessary authentication overhead. These energy-saving strategies enable EMCSABA to consume less energy overall compared to FBESSM and OSCAR, resulting in improved energy efficiency and prolonged network operation.

In a similar fashion the variation of Packet Loss (PL) concerning the simulation time increment has been shown in Fig. 4.7 for the EMCSABA in contrast to FBESSM and OSCAR. FBESSM involves organizing the nodes into clusters, with each cluster having a cluster head responsible for communication with other clusters. This approach reduces the communication overhead and energy consumption in the network. EMCSABA has lower packet loss by 20.33%, 10.45%, 17.31% and 12.36% in comparison to OSCAR under simulation times 50, 100, 150 and 200 seconds respectively. Moreover, packet loss decrement of 18.3%, 7.69%, 14.57% and 10.34% with respect to FBESSM under simulation time 50, 100, 150 and 200 seconds respectively. In Fig. 4.8 the variation of PDR when the simulation time is varied from 50 to 200 seconds has been presented and the comparative analysis of EMCSABA has been done with FBESSM and OSCAR. EMCSABA optimizes channel and slot allocation, reducing collisions and interference, thus enhancing the overall PDR. EMCSABA has a higher PDR value by 6.08 %, 5.63 %, 5.81% and 7.31% in comparison to FBESSM under simulation times 50, 100, 150 and 200 seconds respectively.

Also, in Fig. 4.9 throughput (TP) of EMCSABA is compared with FBESSM and OSCAR by varying the simulation time from 50 to 200 seconds. It is clearly depicted that FBESSM has higher throughput in contrast to OSCAR technique. This is because the combination of fuzzy-based decision-making mechanisms and clustering, in FBESSM can contribute to higher throughput compared to OSCAR, which relies on a different set of mechanisms to manage communication in the network. EMCSABA optimally allocates channels and time slots, minimizing collisions and maximizing available bandwidth, resulting in increased data throughput.

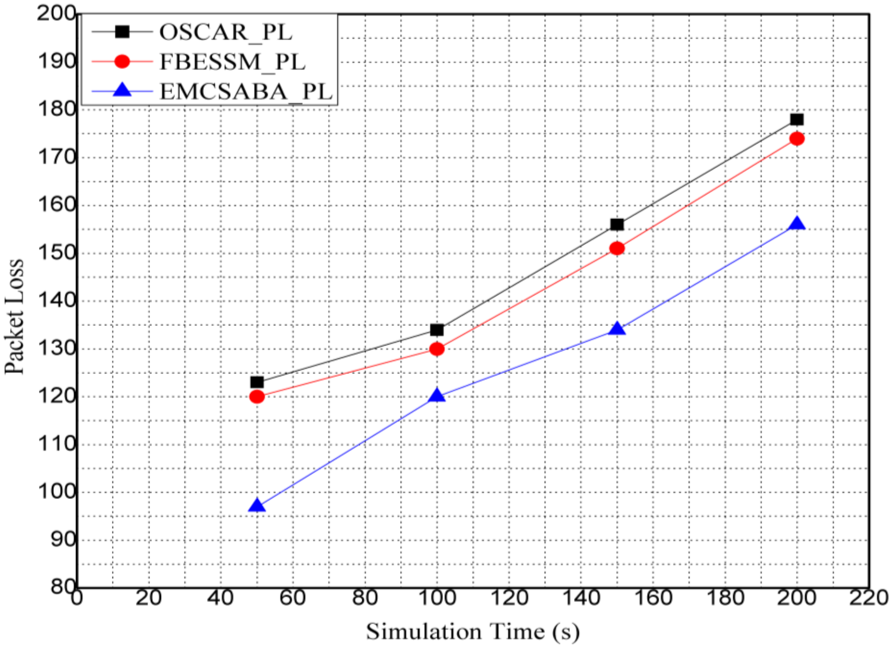


Fig 4.7: Variation of packet loss under the simulation time

Additionally, its belief score authentication reduces authentication overhead, allowing more data to be transmitted within a given time frame. These factors contribute to EMCSABA's higher throughput compared to FBESSM and OSCAR, enabling more efficient and faster data transmission.

The scalability and extended operation of EMCSABA have not been thoroughly investigated, as studies typically concentrate on specified numbers of nodes and durations of simulation. Actual implementation in real-world scenarios may expose difficulties in handling larger networks,increased coordination requirements, and ensuring consistent performance over long durations, which could adversely affect its overall efficiency and dependability.

In order to mitigate the dependence on simulations, it is essential that EMCSABA undergoes thorough field testing in a wide range of real-world settings. This involves implementing it in different network environments, closely monitoring real-time performance measurements, and comparing them with the findings obtained from simulations. Integrating actual data from the real world into the algorithm's refinement process will improve its dependability, resilience, and usefulness in real-life situations. This can be one of the works to be done in the future research.

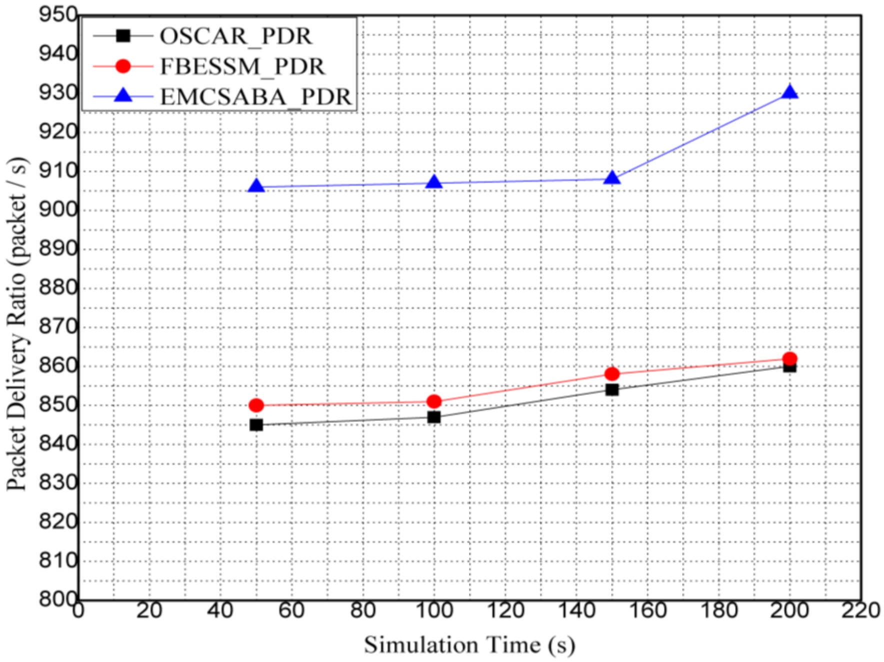


Fig 4.8: Variation of PDR under the simulation time

4.3.3 Comparative Evaluation with Existing Techniques

In this section the comparative evaluation of proposed EMCSABA and FBESSM techniques has been done with other existing approaches such as OSCAR [87], OSCAR- 6 [98], ORCHESTRA-16 [141], TSCH-Orch [142] and e-TSCH-Orch [125] in Table 4.1. It has been inferred from Table 4.1 that the EMCSABA outperforms all other techniques like FBESSM, OSCAR, OSCAR-6, ORCHESTRA-16, TSCH-Orch and e-TSCH-Orch. OSCAR is the least performing one in terms of PDR. Similarly, the evaluation based on EED has also been done in Table 4.2 and it is inferred that the e-TSCH-Orch is the best performing on account of TSCH mechanism involved in the same.

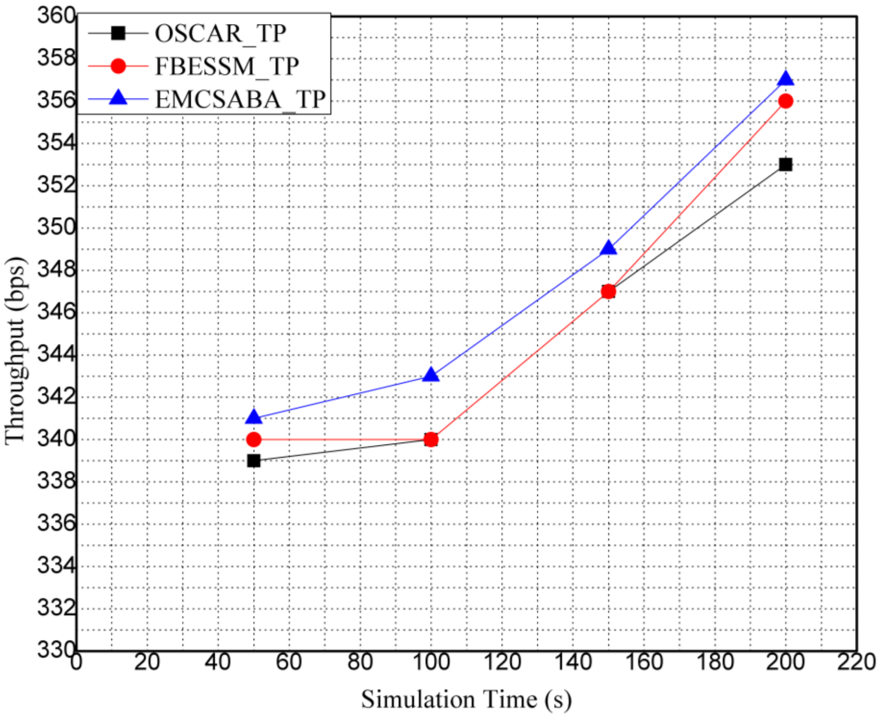


Fig 4.9: Variation of throughput under the simulation time

Table 4.1: Comparative evaluation of EMCSABA with existing techniques based on PDR

Node Count	EMCSABA	FBESSM	OSCAR [87]	OSCAR-6 [98]	ORCHESTRA-16 [141]	TSCH-Orch [142]	e-TSCH-Orch [125]
50	97	96	77	97	95	97	98
100	98	96.5	83	93	90	96	96
150	99	97	86	90	80	96	97
200	99	98	89	88	70	98	96

EMCSABA has potential for a range of real-time applications, particularly in wireless communication systems. EMCSABA has the capability to enhance communication for smart infrastructure in the context of smart cities, ensuring dependable data interchange between sensors and control centers. EMCSABA's effective resource allocation in healthcare can provide advantages for wearable devices and medical sensors by improving data transfer and minimizing latency. EMCSABA can enhance communication in smart factories for industrial IoT, facilitating real-time monitoring and control systems. The inclusion of believe score authentication enhances security, rendering it appropriate for applications that need utmost data integrity, such as banking transactions or the sharing of sensitive information. EMCSABA can be utilized in the field of smart healthcare to enhance connectivity in wearable health devices and remote patient monitoring systems. The optimal distribution of channels and slots, together with believe score authentication, guarantees the safe and timely delivery of vital health data. This program is crucial for delivering healthcare practitioners with up-to-the-minute information and speeding timely intervention.

Table 4.2: Comparative evaluation of EMCSABA with existing techniques based on EED algorithm

Node Count	**EMCSABA**	**FBESSM**	**OSCAR [87]**	**TSCH-Orch [142]**	**e-TSCH-Orch [125]**
50	2.64	2.66	2.76	1.42	0.84
100	2.50	2.52	2.74	1.56	0.92
150	2.42	2.43	2.68	2.64	1.24
200	2.38	2.40	2.62	3.84	1.42

EMCSABA can be utilized in smart cities to enhance communication between different sensors and actuators responsible for monitoring and managing urban infrastructure. These applications include smart traffic management systems, garbage management, and environmental monitoring. The protocol's proficiency in effective resource allocation and secure communication facilitates the requirements of instantaneous decision-making for municipal services, hence enhancing urban living conditions. EMCSABA enables dependable communication across sensors, equipment, and control systems in the realm of industrial automation and the Industrial Internet of Things (IIoT). The ability to limit interference and optimize resource consumption in real-time production environments leads to greater operational efficiency, less downtime, and increased overall productivity.

EMCSABA can be utilized in the field of disaster recovery and emergency management to construct robust communication networks. The protocol's capacity to dynamically assign resources guarantees the dependable transmission of vital information among first responders, emergency agencies, and monitoring systems in real-time, facilitating efficient and unified disaster response measures.

Insufficient discussion on mitigating overheads associated with channel allocation, synchronization, and authentication could significantly impact the practical efficiency of EMCSABA. Efficient channel allocation often requires complex algorithms that can introduce considerable processing overhead, especially in large networks. Synchronization among nodes is crucial to prevent collisions and ensure seamless data transmission, but it can lead to increased latency and coordination overhead. Additionally, implementing robust belief score authentication mechanisms is essential for security but can be computationally intensive, straining resource-constrained nodes. Addressing these overheads requires a careful balance between algorithm complexity, processing efficiency, and resource management to maintain EMCSABA's effectiveness in real-world deployments.

4.4 Summary

Slot allocation in multichannel WSNs for convergecast can be achieved through either centralized or distributed approaches. The centralized scheme involves a central controller assigning slots to nodes, while the distributed scheme allows nodes to collaboratively allocate slots. A new approach named EMCSABA is presented which is a belief score based approach for effective slot allocation in a multichannel environment. Incorporating a belief score authentication scheme strengthens the security of data aggregation, allowing only trusted nodes to participate in the convergecast process. In addition, simulations with varying numbers of nodes and simulation time have shown EMCSABA better performance in contrast to FBESSM and OSCAR. EMCSABA has a higher PDR of 34.7 %, 19.5 %, 16.6 % and 11.23% in comparison to OSCAR under node count 25, 50, 75 and 100 respectively. While in comparison to FBESSM, PDR increments of 1.56 %, 1.87%, 13.96% and 2.04% under node count 25, 50, 75 and 100 respectively. EMCSABA reduces EED by 3.92%, 4.12%, 4.27% and 6.64 % in comparison to FBESSM under simulation time 50, 100, 150 and 200 seconds respectively.

CHAPTER 5

MLMP: MULTICHANNEL LIGHTWEIGHT MAC PROTOCOL

A lightweight multichannel MAC protocol for convergecast WSNs is necessary to address the challenges faced by wireless sensor networks during data gathering. By utilizing multiple channels, the protocol reduces channel contention and collisions, resulting in improved network capacity and data throughput. The efficient allocation of channels and time slots enables concurrent transmissions, reducing latency and ensuring timely data delivery. Additionally, the protocol enhances energy efficiency by incorporating mechanisms such as duty cycling and collision avoidance, thereby prolonging the network's lifespan. In this chapter, a new approach named MLMP is presented which is an efficient way of allocating channels in multichannel environment for Convergecast WSN. The proposed MLMP technique optimizes the performance of convergecast WSNs by maximizing throughput, minimizing latency, and conserving energy.

5.1 Multichannel in Convergecast WSN

Multichannel communication in convergecast wireless sensor networks (WSNs) refers to the utilization of multiple channels for data transmission from multiple sensor nodes to a central sink node. In this context, convergecast refers to the process of gathering data from multiple sensor nodes and aggregating it at a central sink node. The use of multiple channels in convergecast WSNs offers several advantages, including improved network capacity, reduced channel contention, enhanced energy efficiency, and better overall performance. However, certain challenges that needs to be addressed for effective implementation.

5.1.1 Need for Multichannel in Convergecast WSN

Multichannel communication is particularly important in convergecast WSNs due to the following reasons:

- **Increased Network Capacity:** In large-scale WSNs, a single channel may become a bottleneck due to the limited bandwidth. By utilizing multiple channels, the network capacity can be significantly increased, allowing for simultaneous transmissions and reducing congestion.

- **Reduced Channel Contention:** In convergecast scenarios, multiple sensor nodes may attempt to transmit data to the sink node simultaneously. This can lead to collisions and increased packet loss. By using multiple channels, the sensor nodes can transmit concurrently on different channels, reducing contention and improving overall throughput.

- **Improved Energy Efficiency:** In WSNs, energy efficiency is a critical factor since the sensor nodes are typically battery-powered. Multichannel communication enables the allocation of channels in an optimized manner, allowing sensor nodes to transmit and receive data more efficiently. This reduces energy consumption, prolongs the network's lifespan, and enhances overall energy efficiency.

- **Enhanced Reliability:** Multichannel communication provides redundancy and resilience to WSNs. In the case of channel interference or failures, alternative channels can be utilized for data transmission, ensuring reliable communication and minimizing the impact of channel impairments.

5.1.2 Implementation of Multichannel Convergecast in WSNs

The process of implementing multichannel communication in convergecast WSNs involves several key steps:

- **Channel Selection:** The first step is to select the appropriate channels for data transmission. The channels should be carefully chosen to minimize interference and maximize available bandwidth. Factors such as channel availability, frequency band, and interference levels need to be considered. Channels can be statically assigned or dynamically allocated based on network conditions and requirements.

- **Scheduling and Time Slot Assignment:** Once the channels are selected, a scheduling mechanism is established to allocate time slots for each sensor node to transmit their data. The time slots need to be assigned in a coordinated manner to avoid collisions and ensure efficient use of the available channels. Various scheduling algorithms can be employed, such as time-division multiple access (TDMA) or carrier sense multiple access (CSMA) with collision avoidance.

- **Data Transmission:** During their assigned time slots, the sensor nodes transmit their data packets on the allocated channels. Multiple sensor nodes can transmit concurrently on different channels, thereby reducing contention and improving overall network throughput. The sensor nodes sense the channel before transmission to avoid collisions and interference.

- **Receiver Coordination:** The sink node, which acts as the central data collection point, coordinates the reception of data from multiple sensor nodes. It listens on the corresponding channels during the assigned time slots, collecting the data packets sent by the sensor nodes. The sink node may employ techniques such as channel scanning or time synchronization to receive data from different channels efficiently.

- **Convergecast and Data Aggregation:** After receiving the data packets from the sensor nodes, the sink node aggregates the data and performs necessary processing tasks. Convergecast refers to the process of collecting and aggregating data from multiple sensor nodes at the sink node. The sink node may perform data fusion, filtering, or other data processing operations to extract meaningful information from the collected data. Efficient data aggregation techniques should be employed to handle data from multiple channels effectively.

- **Energy Efficiency Considerations:** To conserve energy in multichannel convergecast WSNs, energy-efficient strategies can be employed. Sensor nodes can also dynamically adjust their transmission power based on channel conditions to optimize energy consumption.

- **Collision Avoidance and Interference Mitigation:** To minimize collisions and mitigate interference in multichannel convergecast WSNs, various mechanisms can be employed. These include techniques such as channel sensing before transmission, backoff algorithms to avoid collisions, interference avoidance techniques, and power control mechanisms to mitigate interference.

- **Network Maintenance and Adaptation:** Multichannel convergecast WSNs require ongoing network monitoring and maintenance. As channel conditions change, channel quality and availability need to be periodically evaluated. The network should be able to adapt dynamically by reconfiguring channels, adjusting transmission parameters, or reassigning time slots based on the changing environment.

5.2 Proposed Lightweight Multichannel MAC Scheme

The proposed approach Multichannel Lightweight Mac Protocol (MLMP) involves selecting the channel in the best possible manner. As part of this strategy, the wireless medium is controlled and communicated through timeslots, which are allocated to individual nodes in the network. These time slots are grouped into "frames," and a node may reuse the same timeslot in subsequent frames if there are no conflicts. There are a set number of time intervals in each frame. Any new node must locate a "free" timeslot in which to begin sending data to the network. If a node can’t communicate with its neighbors, then it has to find a timeslot that isn’t being occupied by those neighbors. Also, other nodes whose broadcasts might potentially interfere with or be affected with the node’s transmissions should not utilize the available timeslot. This technique essentially empowers nodes to choose for themselves a period that does not disrupt the communications of other nodes. By assigning each node a specific time slot, one can ensure that messages are sent and received without any collisions or disruptions.

The primary motive of developing MLMP (Multichannel Lightweight MAC Protocol) is to optimize communication efficiency in wireless sensor networks through efficient management of numerous channels. The objective of this protocol is to minimize interference, enhance data throughput, and guarantee low-latency communication. Furthermore, its low weight is specifically intended to save energy and facilitate scaling, making it appropriate for network deployments that have limited resources or are on a vast scale.

5.2.1 Proposed Algorithm for MLMP

Algorithm Multichannel Lightweight MAC protocol (Time slot and Channel Selection)

```
{
Set time slot (periodically a time interval) Timeslot organized into frames
If (no conflict)
Use same timeslot for upcoming frames
If (Discover “free timeslot “) Transmit the data to neighbors
If (is free slot available)
Exchange message with neighbors
}
```

The algorithm begins by setting a predefined timeslot, which represents a specific time interval during which data transmission and reception will occur in the WSN. This timeslot is typically determined based on network requirements and can be periodically repeated. The timeslot is organized into frames, which define the structure of the timeslot interval. Frames help in organizing and synchronizing the data transmission and reception among the sensor nodes. In the first step, the algorithm checks for any conflicts in the timeslot allocation. If there are no nodes, meaning no overlapping timeslots assigned to different sensor nodes, the algorithm proceeds with the same timeslot allocation for upcoming frames. This ensures a consistent and non-conflicting transmission schedule.

If conflicts are detected in the timeslot allocation, the algorithm searches for a "free timeslot." A free timeslot refers to a timeslot that is not currently assigned to any other sensor node for data transmission. Discovering a free timeslot allows the sensor node to avoid contention and collisions during data transmission. Once a free timeslot is discovered, the sensor node can utilize this timeslot to transmit its data to neighboring sensor nodes. This data transmission can be performed using the assigned channel(s) for that particular timeslot. By using a free timeslot, the sensor node can avoid interference and collisions with other concurrent transmissions. Afterwards, it checks if there are additional free slots available for exchanging messages with neighboring sensor nodes. If there are free slots, the sensor node can utilize them to exchange control messages or coordinate information with its neighbors. This facilitates efficient communication and coordination within the WSN.

Let’s illustrate the working of the algorithm with a simple example:

Assume we have a WSN with four sensor nodes (A, B, C, D) and three available channels (C1, C2, C3). The timeslot interval is set to 10 milliseconds, and the timeslots are organized into frames.

1. Frame 1:

- Sensor node A transmits data on channel C1 during timeslot 1.
- Sensor node B transmits data on channel C2 during timeslot 1.
- Sensor node C transmits data on channel C3 during timeslot 1.
- Sensor node D discovers a free timeslot and transmits data on channel C1 during timeslot 2.

2. Frame 2:

- Sensor node A transmits data on channel C1 during timeslot 2.
- Sensor node B transmits data on channel C2 during timeslot 2.
- Sensor node C discovers a free timeslot and transmits data on channel C2 during timeslot 3.
- Sensor node D transmits data on channel C1 during timeslot 2.

3. Frame 3:

- Sensor node A discovers a free timeslot and transmits data on channel C2 during timeslot 3.
- Sensor node B transmits data on channel C2 during timeslot 3.
- Sensor node C transmits data on channel C2 during timeslot 3 (continued).
- Sensor node D discovers a free timeslot and transmits data on channel C3 during timeslot 4.

4. Frame 4:

- Sensor node A transmits data on channel C2 during timeslot 4.
- Sensor node B discovers a free timeslot and transmits data on channel C3 during timeslot 4.
- Sensor node C transmits data on channel C2 during timeslot 4.
- Sensor node D transmits data on channel C3 during timeslot 4.

In this example, each sensor node transmits its data during its assigned timeslot and utilizes the available channels. The algorithm ensures that sensor nodes do not interfere with each other by assigning different timeslots and, if necessary, discovering and utilizing free timeslots. The working of the proposed approach has been depicted in Fig. 5.1.

An efficient MAC protocol is necessary in convergecast WSN to effectively handle communication and data transfer in situations when numerous sensor nodes converge their data towards a central sink node. Convergecast is the act of consolidating information from several source nodes into a single destination node, usually the sink node, in a WSN. In the context of MAC protocols for convergecast WSNs, the phrase "lightweight" refers to a design that aims to minimize communication overhead, energy consumption, and computing complexity. This is especially vital in WSNs, since sensor nodes frequently have limited resources and operate on constrained battery power. The lightweight MAC protocol seeks to achieve a harmonious equilibrium between energy economy and dependable data transmission in convergecast settings.

A lightweight MAC protocol is necessary in WSNs due to the inherent difficulties posed by limited energy resources, communication restrictions, and the scattered nature of sensor nodes. The implementation of a lightweight MAC protocol decreases the communication costs related to channel access, synchronization, and contention resolution. This improvement enhances energy efficiency and extends the lifespan of the network, enabling sensor nodes to function for longer durations without the need for frequent battery changes.

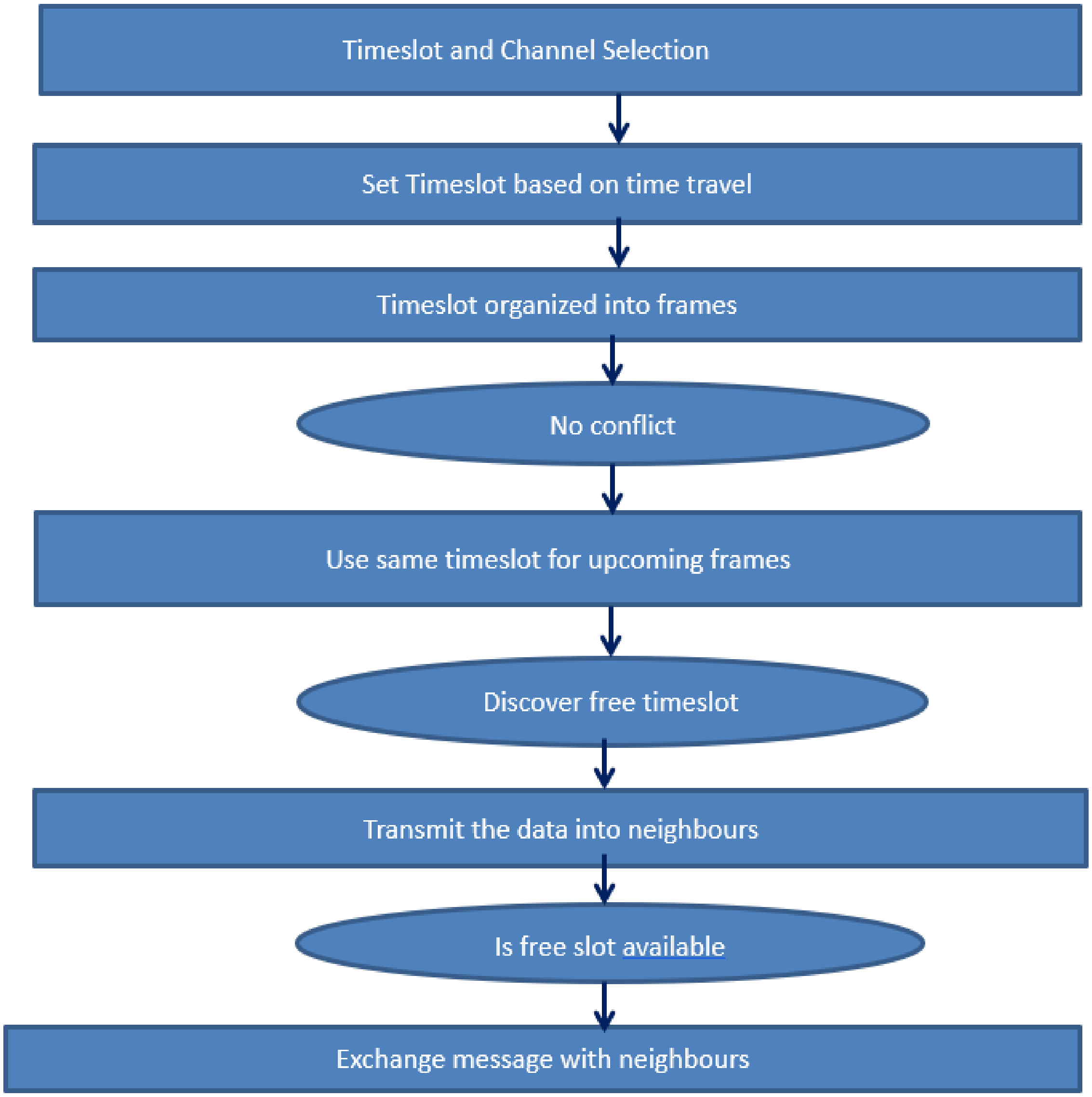


Fig 5.1: Proposed working of lightweight multichannel MAC protocol

In order to overcome the lack of sufficient real-world validation in MLMP, it is necessary to carry out comprehensive field testing in a wide range of situations. Incorporating actual data from the real world into simulations and continuously improving the process would improve the protocol's strength, flexibility, and dependability for practical implementations. But the same has been the part of the future research.

5.3 Comparative Analysis

In this section, the comparative analysis of the proposed MLMP technique with the existing FBESSM and EMCSABA techniques has been done based on metrics like PDR, RE and EC by varying the node count from 25 to 100. MLMP is specifically designed to efficiently manage scalability, even in networks that consist of more than 100 nodes. Through the effective management of various channels, it minimizes conflicts and ensures a high PDR. The lightweight design of the system guarantees minimal energy usage, even when there are more nodes. This is achieved through efficient channel allocation and energy-saving methods, resulting in dependable performance for large-scale deployments. The variation of Energy Consumption (EC) with respect to node count has been shown in Table 5.1.

Table 5.1: EC variation with node count for MLMP

Node count	**EMCSABA_Energy Consumption**	**FBESSM_ Energy Consumption**	**MLMP_ Energy Consumption**
25	10.8	13.8	10.4
50	10	12.2	9.8
75	9.8	11.8	9.5
100	9.3	10.8	9.0

In Fig. 5.2 the variation of EC when the node count is varied from 25 to 100 has been presented and the comparative analysis of MLMP has been done with FBESSM and EMCSABA. It has been depicted that the MLMP technique is performing quite better than the FBESSM and EMCSABA techniques. FBESSM is using the concept of sleep scheduling of nodes that are not participating in the CH selection process and thus reduces the EC to a large extent also EMCSABA is using the belief score mechanism which leads to less consumption of energy owing to non-participation of un-authenticated nodes. But MLMP is performing better on account of the collision-free slot allocation mechanism as adopted which leads to less interference and thus saves more energy. MLP reduces EC by 3.7 %, 2 %, 3.06 %, and 3.22% in comparison to EMCSABA under node count 25, 50, 75, and 100 respectively. Also, MLMP reduces EC by 24.63 %, 19.67 %, 19.94 %, and 16.66 % in comparison to FBESSM under node count 25, 50, 75 and 100 respectively.

From Fig 5.3 it is depicted that the PDR of MLMP is quite high in contrast to FBESSM technique and slightly better in comparison to EMCSABA when the node count is increased from 25 to 100. The reason is that the FBESSM incorporates fuzzy logic-based decision-making mechanisms to optimize energy consumption by controlling the sleep/wake cycle of the nodes whereas MLMP uses collision free slot allocation mechanism which makes more efficient for delivery of packets. MLMP has a slightly higher PDR of 1.03%, 0.50 %, 0,507 % and 0.40% in comparison to EMCSABA under node count 25, 50, 75 and 100 respectively. While in comparison to FBESSM, PDR increments of 2.08 %, 2.07%, 2.05% and 2.06% under node count 25, 50, 75 and 100 respectively. The variation of Packet Delivery Ratio (PDR) with respect to node count has been shown in Table 5.2.

From Fig 5.4 it is depicted that the RE of MLMP is high in contrast to FBESSM technique and slightly better in comparison to EMCSABA when the node count is increased from 25 to 100. MLMP exhibits higher residual energy compared to EMCSABA and FBESSM when varying the node count due to its efficient collision free slot allocation mechanisms. MLMP has a higher RE of 2 %, 1.69 %, 6.25 % and 3.15% in comparison to EMCSABA under node count 25, 50, 75 and 100 respectively. While in comparison to FBESSM, RE increments of 4.08 %, 5.26%, 10.38% and 5.36% under node count 25, 50, 75 and 100 respectively. The variation of Residual Energy (RE) with respect to node count has been shown in Table 5.3.

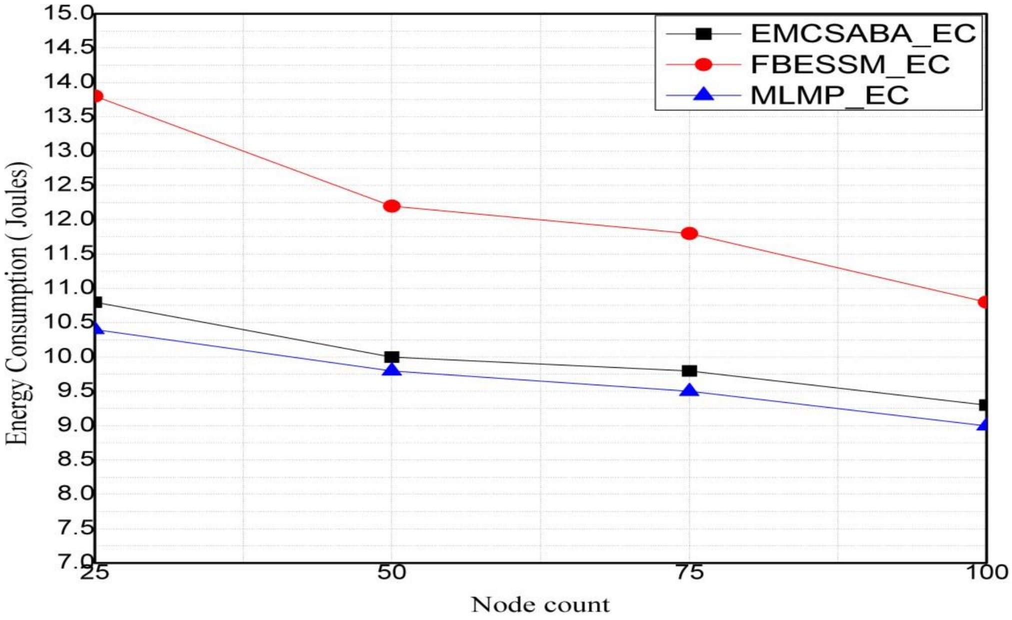


Fig 5.2: EC based comparative analysis by varying the node count

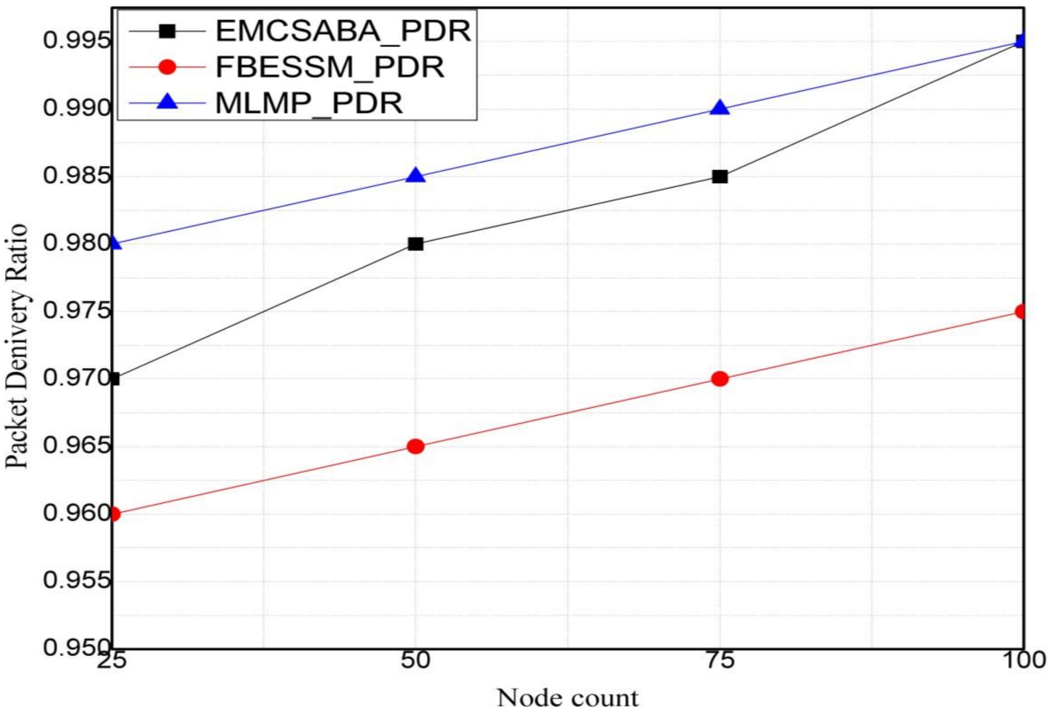


Fig 5.3: PDR based comparative analysis by varying the node count

Table 5.2: PDR variation with node count for MLMP

Node count	**EMCSABA_Packet Delivery Ratio**	**FBESSM_ Packet Delivery Ratio**	**MLMP_ Packet Delivery Ratio**
25	0.97	0.96	0.98
50	0.98	0.965	0.985
75	0.985	0.97	0.99
100	0.995	0.975	0.995

Table 5.3: RE variation with node count for MLMP

Node count	**EMCSABA_Residual Energy**	**FBESSM_ Residual Energy**	**MLMP_ Residual Energy**
25	500	490	510
50	590	570	600
75	800	770	850
100	950	930	980

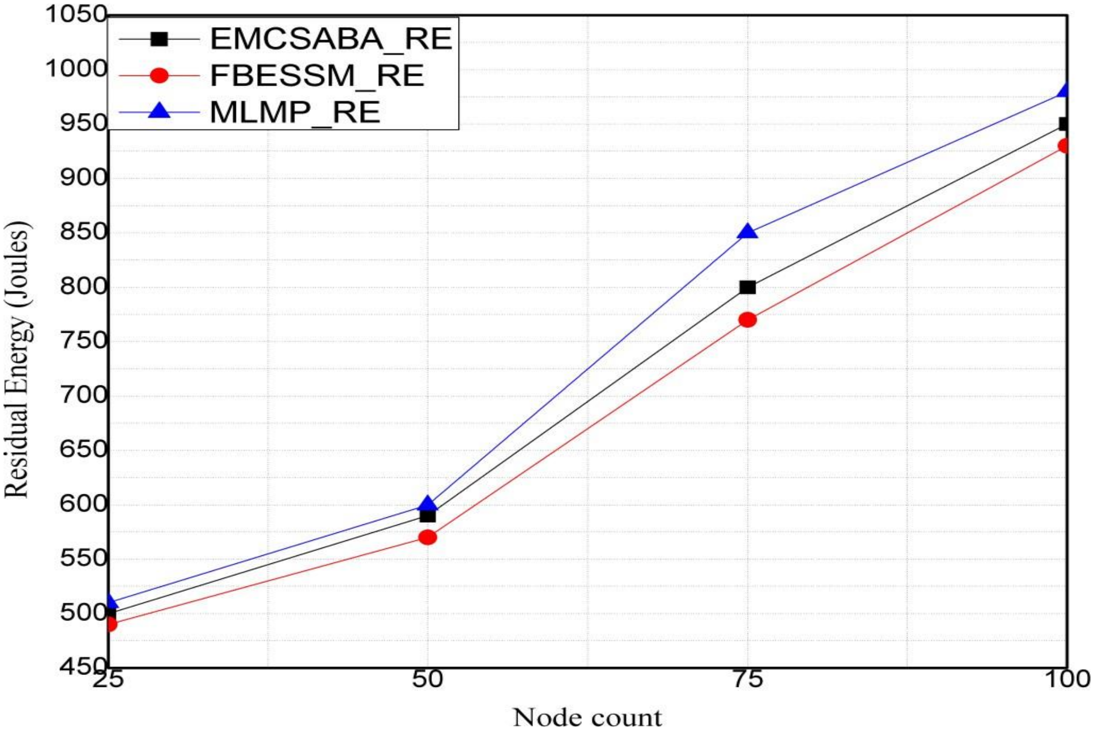


Fig 5.4: RE based comparative analysis by varying the node count

The expected advantages of MLMP encompass a high PDR, decreased latency, and low energy usage resulting from effective multichannel management. Nevertheless, the potential advantages of these theoretical advancements may not consistently result in tangible advantages due to practical obstacles like as interference, synchronization difficulties, and fluctuating network conditions, all of which might affect the actual performance.

MLMP can be utilized in smart retail to enhance communication inside intelligent inventory control systems. Multiple channels enable simultaneous data transfer between goods equipped with RFID tags, payment kiosks, and inventory databases, guaranteeing immediate changes to inventory and preventing instances of exhausted stock. This improves operational efficiency and customer satisfaction by ensuring the accuracy and currency of inventory information. MLMP facilitates effective communication between smart home equipment, hence supporting linked houses and the IoT. The multichannel features of the system allow for simultaneous data sharing across various devices, including smart thermostats, surveillance cameras, and voice-activated assistant. This improves the responsiveness of home automation systems, enabling smooth interactions and prompt implementation of requests in real-time.

MLMP can be utilized in wearable health monitoring devices and patient tracking systems in the healthcare sector. The protocol's capacity to distribute channels in a flexible manner guarantees dependable and minimal delay data transfer, facilitating the real-time monitoring of essential physiological indicators and position tracking. Timely intervention and the provision of up-to-the-minute patient information are crucial for boosting the overall quality of patient care, particularly for healthcare professionals.

MLMP can be employed in education technology to enhance smart classrooms and create interactive learning environments. Multiple channels allow for simultaneous connection between students' devices, interactive whiteboards, and educational servers, enabling immediate collaboration and delivery of materials. This optimizes the learning experience by reducing interruptions and guaranteeing a smooth flow of information in dynamic classroom environments.

The MLMP may encounter drawbacks such as increased complexity in channel administration, probable synchronization challenges across nodes, and increased computing requirements, which might burden devices with limited resources. In addition, it may encounter difficulties in handling scalability in highly congested networks and may not well handle dynamic interference and fluctuating network situations.

5.4 Summary

The process of multichannel convergecast in WSNs involves various processes and one of them is time slot assignment. By carefully addressing the challenges associated with multichannel communication, a well-designed multichannel lightweight MAC protocol for convergecast WSNs can significantly improve network performance, energy efficiency, and data reliability in WSN deployments. The proposed projected algorithm allows nodes in a wireless sensor network to autonomously select non-conflicting timeslots to transmit their data, ensuring communication without interference. Nodes periodically receive timeslots, and if conflicts arise, they discover free timeslots. Frames organize the timeslots, enabling efficient data transmission and preventing interference between nodes and their direct neighbors. It has been derived that MLP reduces EC by 3.7 %, 2 %, 3.06 % and 3.22% in comparison to EMCSABA under node counts 25, 50, 75 and 100 respectively. MLMP has slightly higher PDR of 1.03%, 0.50 %, 0,507 % and 0.40% in comparison to EMCSABA and also MLMP has higher RE of 2 %, 1.69 %, 6.25% and 3.15% in comparison to EMCSABA under node count 25, 50, 75 and 100 respectively.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The conclusions from the analytical and simulation studies conducted for this dissertation are included in this chapter. The goal of this project was to improve a multichannel convergecast based WSN solution that was energy efficient. For this, a fuzzy logic-based energy-efficient technique has first been created, and then a multichannel WSN slot allocation procedure has been designed. In this dissertation, a light-weight multichannel for MAC has been built and demonstrated.

6.1 Conclusion

The process and operation of the convergecast in multichannel WSN have been presented in the first chapter. The various types of multichannel and convergecast WSNs as well as their operational difficulties are addressed. Chapter 2 presents a comprehensive literature review of convergecast networks in terms of MAC protocols, multichannel routing, energy utilization and scheduling methods. The comparative analysis of the techniques has been executed.

In Chapter 3, the FBESSM approach for convergecast WSN has been proposed. It is a fuzzy-based cluster selection technique that is also energy-efficient. There are two main phases to the method. The first stage involves choosing a Cluster Head (CH) based on residual energy and distance using a fuzzy matrix. The data transmission method, which includes control messages, is the second phase. The superior performance of FBESSM on large networks has been demonstrated in simulations with varying numbers of nodes. In comparison to OSCAR, improved network lifetime and packet delivery are the result of increased network capacity and the resulting traffic load.

In Chapter 4, a novel strategy called EMCSABA, which is a belief score-based strategy for efficient slot allocation in multichannel environments, is presented. By limiting participation in the convergecast process to trusted nodes, a belief score authentication scheme improves the security of data aggregation. Additionally, simulations with different node counts and simulation times have demonstrated that EMCSABA performs better than FBESSM and OSCAR. Also, the analysis based on varying node depth also embarks the higher efficiency of EMCSABA with respect to FBESSM and OSCAR.

A well-designed multichannel lightweight MAC protocol for convergecast WSNs has been introduced in Chapter 5 to address the difficulties of multichannel communication and to significantly enhance network performance, energy efficiency, and data reliability in WSN deployments. The suggested algorithm enables nodes in a WSN to autonomously decide which non-overlapping timeslots to use to transmit their data, ensuring unhindered communication. Also, a comparative evaluation of MLMP has been done with EMCSABA and FBESSM.

The complexity of creating and modifying fuzzy logic rules for optimal performance in different network situations may limit FBESSM. Large networks may hinder scalability owing to coordination overhead. EMCSABA must be field-tested in many real-world contexts to reduce simulation dependence. This involves testing it in multiple network contexts, evaluating real-time performance, and comparing results to models. MLMP needs extensive field testing in a variety of environments to overcome its lack of real-world validation. Simulations using real-world data and continual process improvement would strengthen, flexible, and reliable the protocol for practical implementations.

6.2 Future Works

The development of an efficient slot allocation convergecast mechanism for multichannel WSN has been the main focus of this dissertation’s work. Despite the extensiveness of the research work carried out in this dissertation, there is always room for more study on this subject. Following are some areas that need more research:

- Security mechanism for effective data transmission for multichannel communication in convergecast WSN.
- Deployment of machine learning based approaches for efficient slot allocation in convergecast WSN.

- Investigating the impact of node mobility on time slot allocation for optimizing the multichannel lightweight MAC protocol.
- Working on real time dataset and changing environment conditions for the proposed techniques.

Future research initiatives should prioritize thorough real-world experimentation and verification to tackle practical implementation obstacles. It is essential to improve the ability to handle increasing size, strength, and flexibility in response to changing network conditions. Incorporating sophisticated machine learning methods to enhance predictive management and optimize energy efficiency would enhance their performance and dependability in various applications.

References

[1] J. J. Percy and S. Kanthamani, “Revolutionizing wireless communication: A review perspective on design and optimization of RF MEMS switches,” Microelectronics, no. 105891, p. 105891, 2023.

[2] J. Singla, R. Mahajan, and D. Bagai, “An energy-efficient technique for mobile-wireless-sensor-network-based IoT,” Electronics and Telecommunications Research Institute Journal., vol. 44, no. 3, pp. 389–399, 2022.

[3] M. Majid et al., “Applications of wireless sensor networks and Internet of Things frameworks in the industry revolution 4.0: A systematic literature review,” Sensors, vol. 22, no. 6, pp. 2087, 2022.

[4] M. Crocco, M. Cristani, A. Trucco, and V. Murino, “Audio surveillance: A systematic review,” ACM Comput. Surv., vol. 48, no. 4, pp. 1–46, 2016.

[5] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, “Exposure in wireless Ad-Hoc sensor networks,” in Proceedings of the 7th annual international conference on Mobile computing and networking, 2001.

[6] K. F. Haque, A. Abdelgawad, and K. Yelamarthi, “Comprehensive performance analysis of Zigbee communication: An experimental approach with XBee S2C module,” Sensors (Basel), vol. 22, no. 9, p. 3245, 2022.

[7] O. Kanoun, Ed., Energy harvesting for wireless sensor networks: Technology, components and system design. Munich, Germany: De GruyterOldenbourg, 2019.

[8] D. Li, X. Wu, L. Yang, H. Chen, Y. Wang, and X. Zhang, “Energy-efficient dynamic cooperative virtual MIMO based routing protocol in wireless sensor networks,” in 2013 8th International Conference on Communications and Networking in China (CHINACOM), 2013.

[9] K. S. Adu-Manu et al., “WSN architectures for environmental monitoring applications,” J. Sens., vol. 2022, pp. 1–18, 2022.

[10] H. Luo, Z. Huang, and T. Zhu, “A survey on spectrum utilization in wireless sensor networks,” J. Sens., vol. 2015, pp. 1–13, 2015.

[11] T. Jabeen, I. Jabeen, H. Ashraf, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, “An intelligent healthcare system using IoT in wireless sensor network,” Sensors (Basel), vol. 23, no. 11, 2023.

[12] M. Benner and L. Schope, “Using continua health alliance standards - implementation and experiences of IEEE 11073,” in 2011 IEEE 12th International Conference on Mobile Data Management, 2011.

[13] H. Shaban, M. A. El-Nasr, B. Helal, and M. Hamoud, “SportsMBAN: Sports and Tele-healthcare wireless medical body area network for real-time activity monitoring and fall detection,” in IFMBE Proceedings, Cham: Springer International Publishing,

2014, pp. 264–267.

[14] Z. Padrah et al., “An ISA100.11a model implementation for accurate industrial WSN simulation in ns-3,” in 2020 International Workshop on Antenna Technology (iWAT), 2020.

[15] A. Zervopoulos et al.,“Wireless sensor network synchronization for precision agriculture applications,” Agriculture, vol. 10, no. 3, p. 89, 2020.

[16] L. P. Fraile, S. Tsampas, G. Mylonas, and D. Amaxilatis, “A comparative study of LoRa and IEEE 802.15.4-based IoT deployments inside school buildings,” IEEE Access, vol. 8, pp. 160957–160981, 2020.

[17] S. Debdas, S. Chakraborty, B. Biswas, S. Mohapatra, Y. Gupta, and T. Dutta, “Smart Farming using IoT and LoRaWAN,” in 2021 IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), 2021.

[18] Y. Feng, Z. Li, L. Qi, W. Shen, and G. Li, “A compact and miniaturized implanTable antenna for ISM band in wireless cardiac pacemaker system,” Sci. Rep., vol. 12, no. 1, p. 238, 2022.

[19] S. J. Danbatta and A. Varol, “Comparison of zigbee, Z-wave, WI-fi, and Bluetooth wireless technologies used in home automation,” in 2019 7th International Symposium on Digital Forensics and Security (ISDFS), 2019.

[20] N.Baccour, “Radio link quality estimation in wireless sensor networks: a survey,” ACM Transactions on Sensor Networks, vol. 8, no. 4, pp. 34–42, 2012.

[21] W.M. Kempa, “Analytical model of a wireless sensor network (WSN) node operation with a modified threshold-type energy saving mechanism,” Sensors, vol. 19, no. 14, pp. 3114, 2019.

[22] D. K. Bangotra, Y. Singh, N. Kumar, P. Kumar Singh, and A. Ojeniyi, “Energy-efficient and secure opportunistic routing protocol for WSN: Performance analysis with nature-inspired algorithms and its application in biomedical applications,” Biomed Res. Int., vol. 2022, p. 1976694, 2022.

[23] S. C. Ergen and P. Varaiya, “TDMA scheduling algorithms for wireless sensor networks,” Wireless Networks, vol. 16. Springer, 2010.

[24] A. C. Djedouboum, A. A. Ari, A. M. Gueroui, A. Mohamadou, and Z. Aliouat, “Big Data Collection in Large-Scale Wireless Sensor Networks,” Sensors, vol. 18, 2018.

[25] M. K. Singh and S. I. Amin, “Energy efficient wireless sensor network using optimum hops and virtual MIMO technique,” SN Appl. Sci., vol. 2, no. 9, 2020.

[26] S. Kosunalp and Y. Kaya, “IoT-TDMA : A performance evaluation of TDMA scheme for wireless sensor networks with Internet of Things,” Concurr. Comput., vol. 34, no. 21, 2022.

[27] J. Garcia-Morales, G. Femenias, and F. Riera-Palou, “On the design of OFDMA-based FFR-aided irregular cellular networks with shadowing,” IEEE Access, vol. 6, pp.

7641–7653, 2018.

[28] H. Tang, Q. Wang, and L. Hu, “How Network Coding Benefits Converge-Cast in Wireless Sensor Networks,” KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, vol. 7, no. 5, 2013.

[29] R. Soua and P. Minet, “Multichannel assignment protocols in wireless sensor networks: A comprehensive survey,” Pervasive Mob,” Pervasive Mob. Comput, vol. 16, pp. 2–21, 2015.

[30] Q. Huang and Y. Zhang, “Radial coordination for convergecast in wireless sensor networks,” in 29th Annual IEEE International Conference on Local Computer Networks, 2004.[31] V. Sarma Upadhyayula and S. Annamalai, “A low latency and energy-efficient algorithm for convergecast in wireless sensor networks,” in IEEE Global Communications Conference, 2003.

[32] S. Upadhyayula and S. K. S. Gupta, “Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks,” Ad Hoc Netw., vol. 5, no. 5, pp. 626–648, 2007.

[33] H.-W. Tsai and T.-S.Chen, “Minimal time and conflict-free schedule for convergecast in wireless sensor networks,” in 2008 IEEE International Conference on Communications, 2008.

[34] S. Gandham, Y. Zhang, and Q. Huang, Distributed Minimal Time Convergecast Scheduling in Wireless Sensor Networks. Distributed Computing Systems. 2006.

[35] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, “On tree-based convergecasting in wireless sensor networks,” in 2003 IEEE Wireless Communications and Networking, 2003.WCNC 2003, 2004.

[36] S. Gandham, Y. Zhang, and Q. Huang, “Distributed time-optimal scheduling for convergecast in wireless sensor networks,” Computer Networks, Publisher Elsevier, vol. 52, no. 3, 2008.

[37] X. Zhu, B. Tang, and H. Gupta, “Delay Efficient Data Gathering in Sensor Networks,” in Proceedings of the International Conference on Mobile Ad-hoc and Sensor Networks (MSN’05), Wuhan, China, 2005, pp. 380–389.

[38] H. Zhang, P. Soldati, and M. Johansson, “Optimal Link Scheduling and Channel Assignment for Convergecast in Linear Wireless HART Networks,” in Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Piscataway, NJ, USA, 2009.

[39] A. OzlemDurmazIncel, B. Ghosh, and K. K. Krishnamachari, “Multi-Channel Scheduling for Fast Convergecast in Wireless Sensor Networks,” pp. 2008–2009.

[40] G. Tarter, L. Mottola, and G. P. Picco, “Directional antennas for convergecast in wireless sensor networks: Are they a good idea?,” in 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2016.

[41] L. Mottola, T. Voigt, and G. P. Picco, “Electronically-switched directional antennas for wireless sensor networks: A full-stack evaluation,” in 2013 IEEE International Conference on Sensing, Communications and Networking (SECON), 2013.

[42] A. Varshney, T. Voigt, and L. Mottola, “Using directional transmissions and receptions to reduce contention in wireless sensor networks,” in Lecture Notes in Electrical Engineering, Cham: Springer International Publishing, 2014, pp. 205–213.

[43] B. Wei, A. Varshney, W. Hu, N. Patwari, T. Voigt, and C. T. Chou, “DRTI: Directional radio tomographic imaging,” arXiv [cs.OH], 2014.

[44] A. Varshney, L. Mottola, M. Carlsson, and T. Voigt, “Directional transmissions and receptions for high-throughput bulk forwarding in wireless sensor networks,” in Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015.

[45] Y. Wu, J. A. Stankovic, T. He, and S. Lin, “Realistic and efficient multi-channel communications in wireless sensor networks,” in 2008 Proceedings IEEE INFOCOM - The 27th Conference on Computer Communications, 2008.

[46] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, “Flow-based real-time communication in multi-channel wireless sensor networks,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 33–52.

[47] R. Diab and G. Chalhoub, Enhanced multi-channelmac protocol for multi-hop wireless sensor networks”.IEEE Wireless Days, 2014.

[48] O. D. Incel, L. V. Hoesel, P. Jansen, and P. Havinga, “Mc-lmac: a multichannel mac protocol for wireless sensor networks,” Ad Hoc Networks, vol. 9, no. 1, pp. 73–94, 2011.

[49] Y. Kim, H. Shin, and H. Cha, “Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks,” in 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), 2008.

[50] J. Borms, K. Steenhaut, and B. Lemmens, “Low-overhead dynamic multi-channel MAC for wireless sensor networks,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 81–96.

[51] S. S. Bamber, N. Dogra, and M. Angurala, “Modified adaptive mechanism for optimising IEEE 802.15.4 WPANs for wireless sensor networks,” Wirel. Pers. Commun., vol. 128, no. 2, pp. 1391–1415, 2023.

[52] P. Du and G. Roussos, “Adaptive time slotted channel hopping for wireless sensor networks,” IEEE CEEC, pp. 29–34, 2012.

[53] X.-G. Yu, Y.-X. She, B. Li, L. Lei, and S. Lang, “Predictive channel scanning over WLAN,” in 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009.

[54] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, “MMSN:

Multi-frequency media access control for wireless sensor networks," in Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, 2006.

[55] Y. Wu, M. Keally, G. Zhou, and W. Mao, "Traffic-aware channel assignment in wireless sensor networks," in Wireless Algorithms, Systems, and Applications, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 479–488.

[56] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2011, pp. 1–11.

[57] Q. Yu, J. Chen, Y. Sun, Y. Fan, and X. Shen, "Regret matching based channel assignment for wireless sensor networks," in 2010 IEEE International Conference on Communications, 2010.

[58] J. Li, D. Zhang, L. Guo, S. Ji, and Y. Li, "ARM: An asynchronous receiver-initiated multichannel MAC protocol with duty cycling for WSNs," in International Performance Computing and Communications Conference, 2010.

[59] J. Li, X. Guo, and L. Guo, "Joint routing, scheduling and channel assignment in multi-power multi-radio wireless sensor networks," in 30th IEEE International Performance Computing and Communications Conference, 2011.

[60] O. Yilmaz, S. Demirci, Y. Kaymak, S. Ergun, and A. Yildirim, "Shortest hop multipath algorithm for wireless sensor network," Computers and Mathematics with Applications, vol. 63, pp. 48–59, 2012.

[61] Y. Chen, P. H. Gomes, and B. Krishnamachari, "Multi-channel data collection for throughput maximization in wireless sensor networks," in 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, 2014.

[62] R. Diab and G. Chalhoub, "Enhanced multi-channelmac protocol for multi-hop wireless sensor networks," IEEE IFIP WD, pp. 1–6, 2014.

[63] A. Pal and A. Nasipuri, "JRCA: A joint routing and channel assignment scheme for wireless mesh networks," in 30th IEEE International Performance Computing and Communications Conference, 2011.

[64] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, Flow-based real-time communication inmulti-channel wireless sensor networks. Berlin Heidelberg: Springer, 2009.

[65] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in Proceedings of the 1st international conference on Embedded networked sensor systems, 2003.

[66] K. W. Fan, S. Liu, and P. Sinha, "Scalable data aggregation for dynamic events in sensor networks," in Proceedings of ACM 4th international conference on Embedded networked sensor systems, 2006, pp. 181–194.

[67] L. Galluccio, A. T. Campbell, and S. Palazzo, “Concert: aggregation-based congestion control for sensor networks”, In Proceedings of ACM 3rd international conference on Embedded networked sensor systems, pp. 274–275, 2005.

[68] N. R. Junior, M. A. M. Vieira, L. F. M. Vieira, and O. Gnawali, “Data Dissemination Protocol with Network Coding for Wireless Sensor Networks,” in proceedings of the 11th European conference on Wireless sensor networks, 2014.

[69] T. Stathopoulos, J. Heidemann, and D. Estrin, A remote code update mechanism for wireless sensor networks”. 2003.

[70] H. Lee and A. Keshavarzian, “Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks,” in 2008 Proceedings IEEE INFOCOM - The 27th Conference on Computer Communications, 2008.

[71] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, “TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks,” Pervasive Mob. Comput., vol. 5, no. 6, pp. 750–765, 2009.

[72] Y. Wu, J. A. Stankovic, T. He, and S. Lin, “Realistic and efficient multi-channel communications in wireless sensor networks,” in 2008 Proceedings IEEE INFOCOM - The 27th Conference on Computer Communications, 2008.

[73] Y. Bo and J. Li, “Minimum-time aggregation scheduling in multi-sink sensor networks,” in SECON, IEEE, 2011, pp. 422–430.

[74] T. S. Chen, H. W. Tsai, and C. P. Chu, “AdjusTable convergecast tree protocol for wireless sensor networks,” Computer Communications, vol. 33, no. 5, pp. 559–570, 2010.

[75] S. Kumar and P. E. Alice, “Convergecasting in Wireless Sensor Network Through ALBA-P Protocol,” ICESMART, vol. 3.

[76] E. Souza and I. Nikolaidis, “Increasing Aggregation Convergecast Data Collection Frequency through Pipelining,” Wireless Communications and Mobile Computing, 2018.

[77] L. Xu and J. Zhana, “Network Coding Based Converge-Cast Scheme in Wireless Sensor Networks”, Wireless Personal Communications,” An International Journal, vol. 96, no. 4, pp. 4947–4972, 2017.

[78] S. Peter, D. Westhoff, and C. Castelluccia, “A Survey on the Encryption of Convergecast Traffic with In-Network Processing,” IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 1, pp. 20–34, 2008.

[79] H. Shi, M. Zheng, W. Liang, and J. Zhang, “Convergecast Scheduling for Industrial Wireless Sensor Networks with Local Available Channel Sets,” IEEE Sensors Journal, vol. 19, no. 22, pp. 10764–10772, 2019.

[80] Z. Qin, D. Wu, Z. Xiao, B. Fu, and Z. Qin, “Modeling and Analysis of Data Aggregation from Convergecast in Mobile Sensor Networks for Industrial

IoT," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4457–4467, 2018.

[81] M. Bakshi, B. Jaumard, and L. Narayanan, "Optimum ConvergeCast scheduling in wireless sensor networks," IEEE Trans. Commun., vol. 66, no. 11, pp. 5650–5661, 2018.

[82] R. Soua, E. Livolant, and P. Minet, "MUSIKA: A multichannel multi-sink data gathering algorithm in wireless sensor networks," in 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013.

[83] R. Soua, P. Minet, and E. Livolant, "DiSCA: A distributed scheduling for convergecast in multichannel wireless sensor networks," in 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015.

[84] E. N. N. Renato, W. F. Moraes, R. O. Welber, and J. Helder, "Power-efficient and interference-free link scheduling algorithms for connected wireless sensor networks," Wireless Networks, vol. 26, pp. 3099–3118.

[85] Y. Wang and I. Henning, "A deterministic distributed TDMA scheduling algorithm for wireless sensor networks," in 2007 International Conference on Wireless Communications, Networking and Mobile Computing, 2007.

[86] W. L. Lee, A. Datta, and R. Cardell-Oliver, "FlexiTP: A flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks," IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 6, pp. 851–864, 2008.

[87] F.-J. Wu and Y.-C.Tseng, "Distributed wake-up scheduling for data collection in tree-based wireless sensor networks," IEEE Commun.Lett., vol. 13, no. 11, pp. 850–852, 2009.

[88] C.-K. Lin, V. Zadorozhny, P. Krishnamurthy, H.-H.Park, and C.-G.Lee, "A distributed and scalable time slot allocation protocol for wireless sensor networks," IEEE Trans. Mob.Comput., vol. 10, no. 4, pp. 505–518, 2011.

[89]C. Yu, R. Fiske, S. Park, and W. T. Kim, "Many-to-one communication protocol for wireless sensor networks," Int. J. Sens. Netw., vol. 12, no. 3, p. 160, 2012.

[90] M. Bagaa, M. Younis, A. Ouadjaout, and N. Badache, "Efficient multi-path data aggregation scheduling in wireless sensor networks," in 2013 IEEE International Conference on Communications (ICC), 2013.

[91] M. Bagaa, M. Younis, A. Ksentini, and N. Badache, "Multi-path multi-channel data aggregation scheduling in wireless sensor networks," in 2013 IFIP Wireless Days (WD), 2013.[92] B. Zeng and Y. Dong, "A collaboration-based distributed TDMA scheduling algorithm for data collection in wireless sensor networks," J. Netw., vol. 9, no. 9, 2014.

[93] R. Soua, P. Minet, and E. Livolant, "A distributed joint channel and slot assignment for convergecast in wireless sensor networks," in 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), 2014.

[94] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” IEEE Trans. Mob. Comput., vol. 3, no. 4, pp. 366–379, 2004.

[95] J.-M. Kim, S.-H.Park, Y.-J.Han, and T.-M. Chung, “CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks,” in 2008 10th International Conference on Advanced Communication Technology, 2008.

[96] H. Taheri, P. Neamatollahi, O. M. Younis, S. Naghibzadeh, and M. H. Yaghmaee, “An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic,” Ad Hoc Netw., vol. 10, no. 7, pp. 1469–1481, 2012.

[97] S. A. Sert, H. Bagci, and A. Yazici, “MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks,” Appl. Soft Comput., vol. 30, pp. 151–165, 2015.

[98] M. Osman and F. Nabki, “OSCAR: An optimized scheduling cell allocation algorithm for convergecast in IEEE 802.15.4e TSCH networks,” Sensors (Basel), vol. 21, no. 7, p. 2493, 2021.

[99] R. Ghosh, S. Mohanty, P. K. Patnaik, and S. Pramanik, A novel secured method for rapid data accumulation in energy-aware WSN,” in An Interdisciplinary Approach to Modern Network Security. Boca Raton: CRC Press, 2022.

[100] J. John, G. S. Kasbekar, and M. S. Baghini, “Maximum lifetime convergecast tree in wireless sensor networks,” Ad Hoc Netw., vol. 120, no. 102564, p. 102564, 2021.

[101] T. M. Thekkil and N. Prabakaran, “A multi-objective optimization for remote monitoring cost minimization in wireless sensor networks,” Wirel. Pers. Commun., vol. 121, no. 1, pp. 1049–1065, 2021.

[102] A. S. Sadeq, R. Hassan, H. Sallehudin, A. H. M. Aman, and A. H. Ibrahim, “Conceptual framework for future WSN-MAC protocol to achieve energy consumption enhancement,” Sensors (Basel), vol. 22, no. 6, 2022.

[103] S. Saginbekov and A. Jhumka, “Many-to-many data aggregation scheduling in wireless sensor networks with two sinks,” Comput.Netw., vol. 123, pp. 184–199, 2017.

[104] C. Wang, Y. Zhang, and W.-Z.Song, “A new data aggregation technique in multi-sink wireless sensor networks,” in 2014 International Conference on Smart Computing Workshops, 2014.

[105] Y. Huang, C. Zhao, B. Tang, and H. Fu, “Beacon synchronization-based multi-channel with dynamic time slot assignment method of WSNs for mechanical vibration monitoring,” IEEE Sens. J., vol. 22, no. 13, pp. 13659–13667, 2022.

[106] N.-T.Nguyen, B.-H.Liu, V.-T.Pham, and T.-Y.Liou, “An Efficient Minimum-Latency Collision-Free Scheduling 61.Algorithm for Data Aggregation in Wireless Sensor Networks,” IEEE Syst. J, vol. 12, pp. 2214–2225, 2018.

[107] B. Kang, P. K. H. Nguyen, V. Zalyubovskiy, and H. Choo, “A distributed delay-efficient data aggregation scheduling for duty-cycled WSNs,” IEEE Sens. J., vol. 17, no. 11, pp. 3422–3437, 2017.

[108] S. Kumar and H. Kim, “Energy efficient scheduling in wireless sensor networks for periodic data gathering,” IEEE Access, vol. 7, pp. 1–1, 2019.

[109] J. Ma, W. Lou, and X.-Y.Li, “Contiguous link scheduling for data aggregation in wireless sensor networks,” IEEE Trans. Parallel Distrib.Syst., vol. 25, no. 7, pp. 1691–1701, 2014.

[110] B. Yao, H. Gao, Q. Chen, and J. Li, “Energy-adaptive and bottleneck-aware many-to-many communication scheduling for battery-free WSNs,” IEEE Internet Things J., vol. 8, no. 10, pp. 8514–8529, 2021.

[111] M. Bagaa, M. Younis, A. Ksentini, and N. Badache, “Reliable multi-channel scheduling for timely dissemination of aggregated data in wireless sensor networks,” J. Netw.Comput.Appl., vol. 46, pp. 293–304, 2014.

[112] X. Jiao, W. Lou, X. Feng, X. Wang, L. Yang, and G. Chen, “Delay efficient data aggregation scheduling in multi-channel duty-cycled WSNs,” in 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2018.

[113] B. Kang, P. K. H. Nguyen, V. Zalyubovskiy, and H. Choo, “A distributed delay-efficient data aggregation scheduling for duty-cycled WSNs,” IEEE Sens. J., vol. 17, no. 11, pp. 3422–3437, 2017.

[114] Y. Lu, T. Zhang, E.He, and I.-S.Comşa, “Self-learning-based data aggregation scheduling policy in wireless sensor networks,” J. Sens., vol. 2018, pp. 1–12, 2018.

[115] M. Ren, J. Li, L. Guo, X. Li, and W. Fan, “Distributed data aggregation scheduling in multi-channel and multi-power wireless sensor networks,” IEEE Access, vol. 5, pp. 27887–27896, 2017.

[116] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, “Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks,” in 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC), IEEE, 2012, pp. 327–332.

[117] K.-H.Hoi and S.-H.Chung, “A new centralized link scheduling for 6TiSCH wireless industrial networks,” pp. 360–371, 2016.

[118] R. Soua, P. Minet, and E. Livolant, “MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks,” in 2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC), 2012.

[119] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, “Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things,” in 2013 IEEE 14th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), 2013.

[120] R. Soua, P. Minet, and E. Livolant, “Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15. 4e TSCH networks,” Transactions on Emerging Telecommunications Technologies, vol. 27, pp. 557–575, 2016.

[121] A. Aijaz and U. Raza, “A decentralized adaptive multi-hop scheduling protocol for 6TiSCH wireless networks,” IEEE Sens. J., vol. 17, no. 20, pp. 6825–6836, 2017.

[122] Demir, A. K., &Bilgili, S, “DIVA: a distributed divergecast scheduling algorithm for IEEE 802.15. 4e TSCH networks”. Wireless Networks, 25(2), 625–635, 2019.

[123] S. Duquennoy, B. Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust mesh networks through autonomously scheduled TSCH,” in Proceedings of the 13th ACM conference on embedded networked sensor systems, 2015, pp. 337–350.

[124] S. Oh, D. Hwang, K.-H. Kim, and K. Kim, “Escalator: An autonomous scheduling scheme for convergecast in TSCH,” Sensors (Basel), vol. 18, no. 4, 2018.

[125] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, and L. A. Grieco, “Autonomous and traffic-aware scheduling for TSCH networks,” Comput.Netw., vol. 135, pp. 201–212, 2018.

[126] Y. Wang and I. Henning, “A deterministic distributed TDMA scheduling algorithm for wireless sensor networks,” in 2007 International Conference on Wireless Communications, Networking and Mobile Computing, 2007.

[127] W. L. Lee, A. Datta, and R. Cardell-Oliver, “FlexiTP: A flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 6, pp. 851–864, 2008.

[128] C.-K.Lin, V. Zadorozhny, P. Krishnamurthy, H.-H.Park, and C.-G.Lee, “A distributed and scalable time slot allocation protocol for wireless sensor networks,” IEEE Trans. Mob.Comput., vol. 10, no. 4, pp. 505–518, 2011.

[129] M. Meng, Z. Yujun, Z. Dadong, and C. Fan, “Scheduling for data transmission in multi-hop ieee 80215,” Mobile Networks and Applications, vol. 23, pp. 119–125, 2018.

[130] K.-H.Phung, B. Lemmens, M. Goossens, A. Nowe, L. Tran, and K. Steenhaut, “Schedulebased multi-channel communication in wireless sensor networks: A complete design and performance evaluation,” Ad Hoc Networks, vol. 26, pp. 88–102, 2015.

[131] G. Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, and N. Montavont, “Leapfrog collaboration: Toward determinism and predictability in industrial-IoT applications,” in 2017 IEEE International Conference on Communications (ICC), 2017.

[132] S. Kim, H.-S. Kim, and C. Kim, “Alice: autonomous link-based cell scheduling for tsch,” in Proceedings of the 18th International Conference on Information Processing in Sensor Networks, 2019, pp. 121–132.

[133] A. Karalis, D. Zorbas, and C. Douligeris, “Collision-free advertisement scheduling for IEEE 802.15.4-TSCH networks,” Sensors (Basel), vol. 19, no. 8, p. 1789, 2019

[134] O. Tavallaie, J. Taheri, and A. Y. Zomaya, “Design and optimization of trafc-aware tsch scheduling for mobile 6tisch networks,” in Proceedings of the International Conference on Internet-of Things Design and Implementation, 2021, pp. 234–246.

[135] B. K. Sen, S. Khatua, and R. K. Das, “Multichannel Pipelined Scheduling for Raw Data Convergecast in Sensor-Cloud,” *Mobile Networks and Applications*, pp. 1–17, 2023.

[136] M. Kherbache, O. Sobirov, M. Maimour, E. Rondeau, and A. Benyahia, “Decentralized TSCH scheduling protocols and heterogeneous traffic: Overview and performance evaluation,” *Internet of Things*, vol. 22, no. 100696, p. 100696, 2023.

[137] J. Liu, L. Zhang, F. He, C. Zhang, S. Jiang, and X.-Y. Li, “Communication-efficient regret-optimal distributed online convex optimization,” *IEEE Trans. Parallel Distrib. Syst.*, pp. 1–13, 2024.

[138] M. G. Gaitán, L. Almeida, P. M. Dorey, P. M. Santos, and T. Watteyne, “Minimal-Overlap Centrality for Multi-Gateway Designation in Real-Time TSCH Networks,” *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 1, pp. 1–17, 2024.

[139] A. Hkiri, M. Karmani, and M. Machhout, “The Routing Protocol for low power and lossy networks (RPL) under Attack: Simulation and Analysis,” in 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), 2022.

[140] B. A. Mohamed, B. A. Abdelhakim, S. Sohaib, F. Hassan, and E. B. Soumaya, “Comparative study on the density and velocity of AODV and DSDV protocols using OMNET++,” in Lecture Notes on Data Engineering and Communications Technologies, Cham: Springer International Publishing, 2023, pp. 176–189.

[141] S. Duquennoy, B. Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH,” in Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems-SenSys ’15, Seoul, Korea, 2015, pp. 337–350.

[142] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust mesh networks through autonomously scheduled TSCH,” in Proceedings of the13th ACM Conference on Embedded Networked Sensor Systems, 2015.