CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Partition Identification and Reconnectivity in Wireless Sensor Networks

by

Kashif Nasr

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Computing
Department of Computer Science

2023

Partition Identification and Reconnectivity in Wireless Sensor Networks

By

Kashif Nasr

(PC103008)

Dr. Feng Xia, Professor

RMIT University, Australia

(Foreign Evaluator 1)

Dr. Muhammad Imran, Professor

University of Glasgow, UK

(Foreign Evaluator 2)

Dr. Noor Muhammad Khan

(Thesis Supervisor)

Dr. Abdul Basit Siddiqui

(Head, Department of Computer Science)

Dr. Muhammad Abdul Qadir

(Dean, Faculty of Computing)

DEPARTMENT OF COMPUTER SCIENCE

CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

ISLAMABAD

2023

Dedicated to my Parents and wife who always back me up in hard times

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD
Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled "**Partition Identification and Re-connectivity in Wireless Sensor Networks**" was conducted under the supervision of **Dr. Noor Muhammad Khan**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Science, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Computer Science.** The open defence of the thesis was conducted on **June 19, 2023**.

Student Name : Kashif Nasr (PC103008)

The Examination Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a)	External Examiner 1:	Dr. Adeel Akram, Professor UET, Taxila
(b)	External Examiner 2:	Dr. Tauseef Jamal, Professor PIEAS, Islamabad
(c)	Internal Examiner :	Dr. Muhammad Siraj Rathore Assistant Professor CUST, Islamabad

Supervisor Name : Dr. Noor Muhammad Khan
Professor
CUST, Islamabad

Name of HoD : Dr. Abdul Basit Siddiqui
Associate Professor
CUST, Islamabad

Name of Dean : Dr. Muhammad Abdul Qadir
Professor
CUST, Islamabad

AUTHOR'S DECLARATION

I, **Kashif Nasr (Registration No. PC103008)**, hereby state that my PhD thesis titled, '**Partition Identification and Re-connectivity in Wireless Sensor Networks'** is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.

(Kashif Nasr)

Dated: June, 2023

Registration No: PC103008

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Partition Identification and Re-connectivity in Wireless Sensor Networks**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.

(**Kashif Nasr**)

Dated: June, 2023 Registration No: PC103008

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

1. **Kashif Nasr**, Noor M. Khan "Toward connectivity of a disconnected cluster in partitioned wireless sensor network for time-critical data collection," *International Journal of Distributed Sensor Networks*, vol. 16, no. 12, 2020. DOI: https://doi.org/10.1177/1550147720984654

(Kashif Nasr)

Registration No: PC103008

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Prof. Dr. Noor. M. Khan for the continuous support during my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me throughout this research and writing of this thesis. I could not have imagined a better advisor and mentor for my Ph.D. Beside my advisor, I would like to thank my Parents for their emotional support, whenever i was stuck or stressed during my Ph.D. I am also very thankful to the cooperation of the Dean, Faculty of computing and the University management, Whose co-operation enabled me to conclude my PhD.

(Kashif Nasr)

Abstract

A wireless sensor network (WSN) is a kind of multi-hop network that is widely used for monitoring and surveillance purposes. The sensors are deployed over a large area and the basic objective is to sense their environment and communicate useful information to a central repository for further processing and decision-making. Thus, all the sensors need to be connected to that central repository, called a sink, either directly or through some intermediate nodes. The failure of certain sensor nodes located at critical positions creates a situation in which some alive nodes are deprived of their communication to the sink. This network partitioning phenomenon may render a large number of alive nodes useless by preventing them from sending important time-critical data to the sink. This problem consists of two components, namely,(1) Partition identification and (2) Re-connectivity.

This dissertation presents three different solutions to this problem. The first solution named as Partition Identification and Redeployment Algorithm (PIRA) is based on a traditional network topological information-based approach, where the sink node identifies the occurrence of partitions and proposes an appropriate redeployment based on its knowledge of the network topology and the state of the sensor nodes. The second proposed solution is a novel cooperative beamforming-based approach. The third solution is based on human-assisted sensor node energy visualization that involves vulnerable nodes identification and optimized redeployment decisions.

The Proposed PIRA employs a two-step approach. In the first step, network partitions caused by the failure of key sensor nodes that ensure connectivity are identified. In the second step, a potential location for the optimized redeployment of additional relay nodes is identified in order to restore connectivity of the isolated partitions, with the rest of the network. Received Signal Strength Indicator(RSSI) is used to figure out the effective communication range of the relay nodes to account for the effect of obstacles and other objects blocking the line of sight between sensor nodes. This helps in determining where to put more relay nodes over irregular terrain. When compared with the existing techniques in terms

of the number of partitions detected, the duration of partition detection, and the energy consumption of the sensor nodes, the proposed technique not only detects network partitioning in the shortest possible time without any additional message congestion but also leads to a significant reduction in energy consumption.

In the second solution, a novel partition discovery and re-connectivity mechanism are proposed which exploits a cooperative beamforming approach. This method allows a group of sensor nodes trapped in an isolated partition to cooperate and create a directional beam that significantly extends their overall communication range to restore communication by connecting with a remote relay node connected to the core network. The proposed mechanism lets the disconnected sensor nodes trapped in an isolated partition be discovered by the sink while restoring their connectivity without requiring additional resources. The proposed method considered partitioning scenarios that occur due to node failures during deployment as well as network operation. Extensive computer simulations show that the proposed mechanism performs partition detection and restores connectivity of the wireless sensor network in a timely manner. It achieves up to 70% Partition healing in near real-time, making it a viable solution for real-time Wireless Sensor Network applications.

The third proposed solution relies on the visualization of the energy state information of sensor nodes in order to detect their vulnerability for their prospective failure. It facilitates the network manager to predict and prevent possible network fragmentation. It is also helpfull in assissting the network manager to make redeployment decisions, by identifying the exact location and instance to redeploy additional sensor nodes to ensure connectivity. The proposed visualization technique is subjected to evaluation by ten experts, and their feedback is used to calculate performance metrics including precision, recall, and accuracy. The experts' feedback indicates that the proposed visualization method enables the network managers to detect vulenrable nodes with an accurcay rate exceeding 90%.

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analoge to Digital Converter
AOI	Area of Interest
DPDRU	Distributed Partition Detection and Recovery using UAV
GPS	Global Positioning System
HCCI	Hop Count based Cut detection algorithm
IDCRWSN	Intelligent on-Demand Connectivity Restoration for Wireless Sensor Networks
IoT	Internet of Things
LOS	Line of Sight
MEMS	Micro Electromechanical System
MIMO	Multiple-Input Multiple-Output
NLOS	Non Line of Sight
PIRA	Partition Identification and Redeployment Algorithm
PIRCB	Partition identification and Reconnectivity through Cooperative Beamforming
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time
VANET	Vehicular Ad-hoc Network
VSA	Virtual Smart Antenna system
WSN	Wireless Sensor Network

Symbols

V	Total number of sensor nodes
V_s	List of alive connected nodes
V_p	List of partitioned nodes
$\Delta\phi$	Phase difference
$\widetilde{G}$	Isolated partition graph
P_i	List of Isolated partitions
d_{ij}^{est},	Distance estimated through RSSI,
d_{ij}^{calc}	Distance calculated through Euclidean distance

Chapter 1

Introduction

This chapter begins with an overview of the wireless sensor network in Section 1.1, Sensor nodes and their critical components are examined in Section 1.2. The architecture and types of wireless sensor networks are discussed in Section 1.4 and 1.3. Whereas section 1.5 discusses numerous real-world applications for wireless sensor networks, while Section 1.6 discusses some critical design challenges. Section 1.8 discusses the importance of connectivity, while Section 1.9 discusses network partitioning. Section 1.10 discusses beamforming and the rationale for using collaborative beamforming. whereas, section 1.11 discusses the visualisation and its importance. The research objectives and Thesis Organization are discussed in Section 1.7 and 1.12, respectively.

1.1 Wireless Sensor Networks

Advances in microelectromechanical systems (MEMS) have ushered into an era of intelligent electronic systems in the form of wireless sensor networks (WSNs). In general, a wireless sensor network is made up of sensor nodes, an area of interest, and one or more sink nodes. The area of interest or target is usually a physical object, location, or phenomenon within the environment that produces the observed conditions. A sensor node is required to continuously monitor these targets and collect the relevent data. After retrieving the data, the sensor nodes have the task of communicating it to the sink node. To achieve this task, the sensor node and the sink node must be connected through specific communication channels. A sensor node can be connected directly to a sink node or through numerous intermediate relay nodes, as shown in the Figure. 1.1.

This allows the nodes to establish an ad hoc network and transmit messages across multiple hops in a remote and hostile environment. Due to their small size, sensors have some fundamental limitations such as energy, limited computing power, and memory size [1]. Although advances in electronics and wireless communication technology have made it possible to overcome these fundamental limitations to some extent, there is still a need for optimization to extend the application range of WSN [2, 3].

WSNs have been widely deployed in recent years for surveillance and monitoring purposes, such as monitoring human health and habitats, tracking enemy troops and equipment, assessing the structural integrity of buildings and bridges, and recording and regulating natural disasters such as earthquakes, volcanoes, forest fires, floods, and storms [4].

In addition, the Internet of Things (IoT), being a variant of WSN and a network of interconnected computing devices equipped with a variety of sensors, has become one of the most important technologies in daily life and will likely define the future of humanity. These tiny sensor devices are the backbone of any basic IoT system. They collect raw data from a given environment and transmit it to a processing unit. The base layer of the IoT architecture is responsible for the connectivity of the sensors and data collection over a network [5].

Network connectivity plays an important role in the overall efficiency of a WSN network. All the raw data collected by the sensors needs to be transmitted to the sink for further processing. Therefore, it is necessary to ensure that each sensor node has an active path to the sink, either directly or through an intermediate relay node. Due to the nature of multi-hop wireless communications, the intermediate relay nodes play an important role in maintaining overall network connectivity. If these intermediate relay nodes fail due to energy depletion or any other hardware malfunction, it will cause the network partitioning. As a result, there is a high probability that a group of sensor nodes relying on these intermediate relay nodes will lose communication with the sink, depriving them of the ability to send collected data to the sink. A conventional solution to the problem is to replace the faulty node and restore the connection, which is a time-consuming task, and most likely the data collected by the sensor during this time will lose its significance, or it simply cannot store large amounts of data due to memory limitations. The situation becomes even more critical when the collected data is of greater importance and is subject to time constraints. This is the case, for example, in WSN applications for healthcare and real-time monitoring. Therefore, in such scenarios, we need a robust, fault-tolerant mechanism that restores connectivity in real time and provides a path for time-critical, important data to reach the sink.

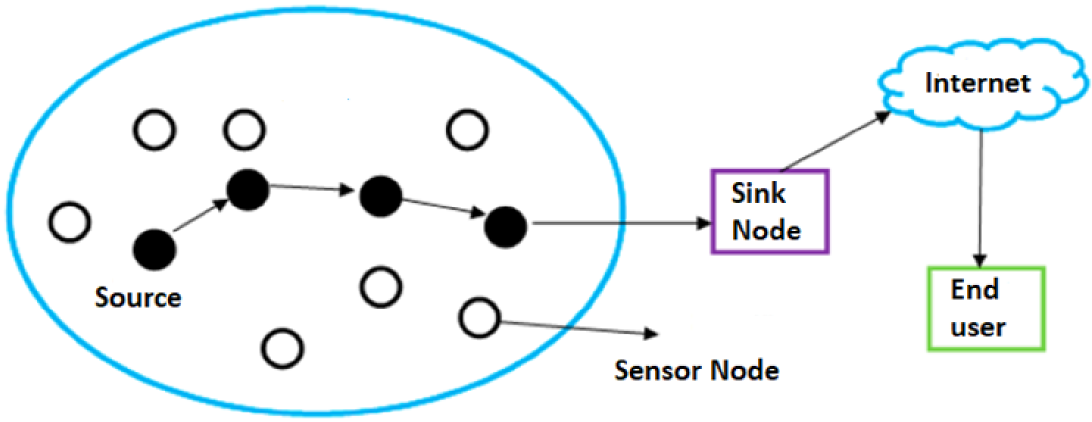

FIGURE 1.1: Wireless Sensor Network

1.2 Wireless Sensor Node

A typical wireless sensor node consists of four main modules, the sensing unit, the processing unit, the memory unit and the communication unit as shown in Figure 1.2. These fundamental components form the backbone of wireless sensor networks (WSNs), providing the essential functionalities for data detection, communication, processing, and storage. The sensor unit serves as the primary element for detecting events or phenomena within the area of interest, while the communication unit facilitates seamless information exchange between the nodes and the sink. To enable effective data analysis and decision-making, sensor nodes also require a processing unit for pre-processing the detected data and a memory unit for storing received data. The interplay of these components, as illustrated in Figure 1.2, enables the successful operation of WSNs and facilitates the collection and dissemination of valuable information.

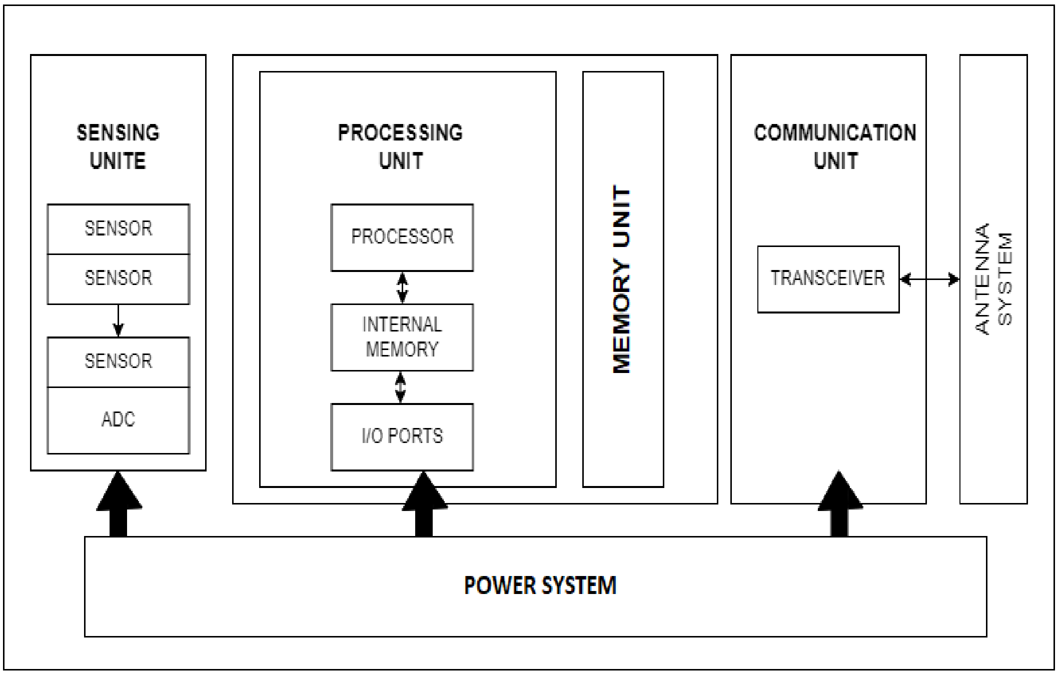

FIGURE 1.2: Sensor Node

1.2.1 Sensing Unit

This entity is responsible for collecting data from its surrounding environment. They respond to a physical stimulus such as temperature, pressure, humidity, etc. through a signal and response mechanism. Sensor nodes are often classified into two types based on their operating principle: active sensor nodes and passive sensor nodes.[6].

Passive sensor nodes rely on the natural energy emitted by the source for their operation; they observe the target of interest using the energy/signals emitted by the source. They do not generate their own signal and instead rely on the signal from the source to detect them. Active sensors contrast with passive sensors in that they actively emit signals towards the target of interest for detection, generating their own echo signals. This active approach require a dedicated power source, making active sensors more power-intensive compared to passive ones. As a result, active sensor systems tend to consume more power, emphasizing the need for efficient power management strategies in order to prolong their operational lifespan.

The area around the sensor node in which it can detect any event that occurs is called sensing region. So, the range of a sensor node is the maximum distance between the node and the place where the target event can be found in the sensing area.

1.2.2 Processing Unit

A central processing unit is an essential component of any sensor node; it is responsible for all computational and regulatory operations within the sensor node. The capabilities of a processing unit have a significant impact on a node's energy consumption and computational capacity. A microcontroller and associated peripherals such as memory, analog-to-digital converters (ADCs), and interfaces form a complete processing unit [7]. Several companies produce microcontrollers for low-power systems such as sensor networks, there are usually three types of microcontrollers that are widely used in wireless sensor networks. They are Microchips (PIC Microcontrollers), the AVR series and the MSP430 series. Feature comparison of a few widely used processing units are given in the table 1.1.

1.2.3 Memory Unit

A memory unit is a type of storage device that is used to store data collected by the sensors or other information that is needed for the operation of the node. Since the sensor node is an energy constraint device, it is important to optimize

Table 1.1: Comparative analysis of sensor nodes

Parts/Sensor	Mica	Telos/Tmote	Enocean	iMote2
Processing Unit	ATMega128 (AVR Series)	TiMSP430F149 (MSP430 Serires)	PIC18F452 (PIC Micro Controllers)	Inter PXA271
Manufacturer	ATMEL	Texas Instruments	Microchip	Intel
Processing Speed	4MHz	8MHz	40MHz	13MHz to 400MHz
Memory	128KB	69KB Flash, 2Kb RAM	32Kb Flash, 1.5 Kb RAM	32MB Flash, 32MB SDRAM
Power	2.7V -5.5 V	2.1V-5.5V	2V - 5.5V	3.2 V-4.5V
Interfaces	UART	GPIO,UART	UART,SPI, 1 2C	3xUART,2xSPI I2C,SDIO
Bus/Bit	8 bit	12 bit	8 bit	32bit
Analogue Support	10 bit ACD	10 bit ADC	10 bit ADC	Nil

its memory consumption and management. For this reason, there are three basic types of memory used in a sensor unit in terms of energy consumption, namely an integrated microcontroller memory, a flash memory for application program storage, and a random access memory (RAM) for efficient data retrieval and storage. [8]. The size of the memory unit may vary depending on the specific requirements of the sensor node and the amount of data that needs to be stored.

1.2.4 Communication Unit

This unit is responsible for the communication among sensor nodes. It enables the sensor unit to communicate control and data packets to be received and transmitted to the required destination. Generally, ”communication range” refers to the area around a sensor node in which other sensors can receive its signals. The normal simplified form of communication range representation can be a circular disk around the node, where the radius of the disk defines the communication range as

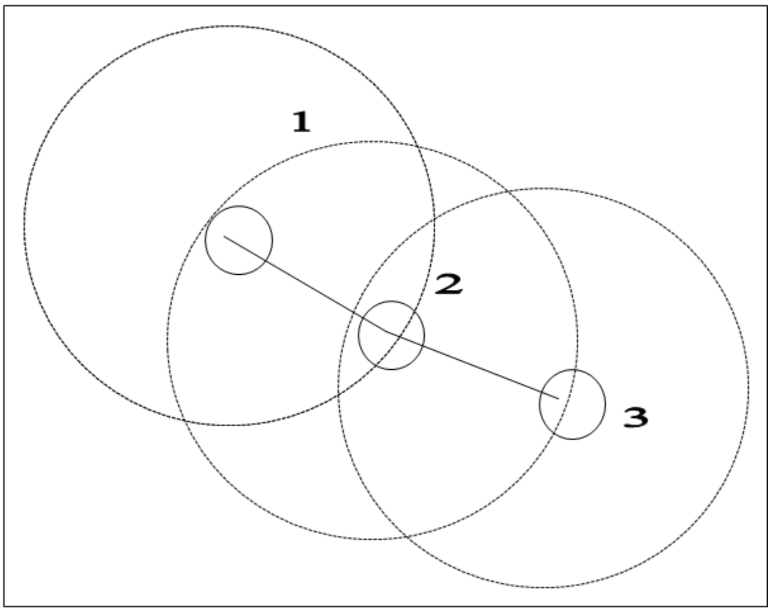

FIGURE 1.3: Disk Communication Range

show in Figure 1.3. However, there are several factors such as fading, scattering, blockages in the communication path [9] that affect the communication range and result in an irregular disk [10, 11] as shown in Figure.1.4. Sensor nodes and sinks communicate using wireless channels, such as the Radio frequency (RF) channel. The RF channel is an electromagnetic wave used to transmit signals containing data. The most important characteristics of RF communication devices are antenna size and power usage. In addition, the type of modulation, data rate, and transmission power also affect the performance of an RF radio.

1.3 Types of Wireless Sensor Networks

In general, wireless sensor networks are very application specific and the characteristics of one wireless sensor network are normally different from those of another wireless sensor network. Therefore, they are deployed according to the application's requirements and needs. Some of the major types of wireless sensor networks that are generally deployed to achieve required application goals are as follows:

- Static and Mobile Wireless Sensor Networks

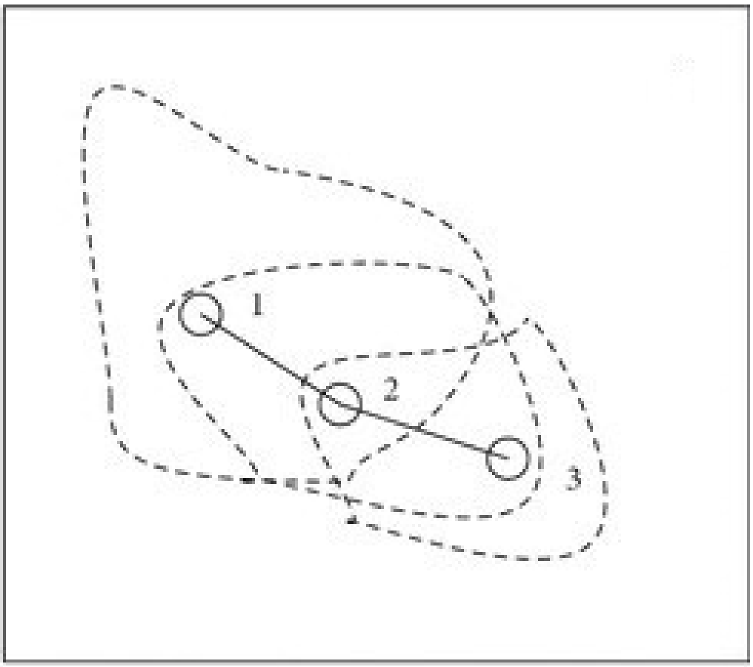

FIGURE 1.4: Irregular Communication Range

- Deterministic and Non-Deterministic Wireless Sensor Networks
- Single-sink and Multiple-sink Wireless Sensor Networks
- Mobile-sink and Static-sink Wireless Sensor Networks
- Homogeneous and Heterogeneous Wireless Sensor Networks
- Self-configurable and Non Self-configurable Wireless Sensor Networks

1.3.1 Static and Mobile Wireless Sensor Networks

A static wireless sensor network is a network in which the sensors and sink are fixed in a specific location and do not move. A static WSN is suitable for applications that require a stable and reliable connection between the sensors and the sink, and that do not require the sink to move. Examples of static WSNs include monitoring systems in hospitals, factories, or buildings, or environmental monitoring systems in natural reserves or remote areas. A mobile wireless sensor network is a network in which at least one of the sensors or the sink is equipped with mobility and can move within the coverage area of the sensors. A mobile WSN is suitable for applications that require the sink to move and cover a larger area, or that require the sink to be closer to the sensors for a better signal quality. Examples of mobile WSNs include healthcare systems that monitor the vital signs of patients as they move, agriculture systems that optimize the irrigation of crops based on the real-time soil moisture and weather conditions, or security systems that detect and prevent intrusions in perimeter fences or protected areas.

The choice between a static WSN and a mobile WSN depends on the specific requirements and constraints of the application and the environment. A static WSN may be more suitable for applications that require a stable and reliable connection between the sensors and the sink, and that do not require the sensor nodes to move around. A mobile WSN may be more suitable for applications where one needs to monitor some moving objects, such as animals monitoring require mobile sensor nodes. Static sensor nodes are simple, energy-efficient, and easy to manage. Whereas mobile sensor networks are complex and difficult to manage, due to their variable topology, which changes over time, making it difficult for the routing protocol to perform efficient communication of data to the sink.

1.3.2 Deterministic and non-Deterministic Wireless Sensor Networks

The deployment of deterministic and non-deterministic wireless sensor networks may differ depending on the specific requirements and constraints of the application. In the case of a deterministic wireless sensor network, the deployment may involve carefully planning the placement and configuration of the nodes in the network to ensure that the network behaves in a predictable and reliable manner. This may involve considering factors such as the coverage area, the transmission range of the nodes, and the power consumption of the nodes. Wireless sensor networks are generally deployed in an unattended harsh environment, thus a deterministic deployment of sensor nodes is very difficult to perform, where the location of the sensor nodes is identified and fixed. Although pre-planned deterministic wireless sensor networks are very efficient and cost-effective. In general, a very limited application have the privilege of performing deterministic sensor node deployment.

On the other hand, non-deterministic wireless sensor where the location of the wireless sensor nodes is not pre-calculated and fixed, rather the sensor nodes are randomly deployed in the area of interest with certain density. Although, these kinds of networks are easy to deploy and usually work in most of the situations, but they do not provide a cost-effective solution. They mostly suffer from connectivity and coverage issues and require complex control system.

1.3.3 Single-sink and Multiple-sink Wireless Sensor Networks

A single sink wireless sensor network is a WSN in which all the sensors transmit their data to a single central sink, which is responsible for collecting, storing, and processing the data. A single sink WSN is suitable for applications that do not have stringent requirements for data collection and processing, and that do not need to support a large number of sensors or a large volume of data. A single sink WSN is also simpler and easier to design, deploy, and maintain, as it only requires a single sink and a single communication protocol.

A multiple sink wireless sensor network is a network in which the sensors transmit their data to multiple sinks, which are distributed throughout the coverage area of the sensors. A multiple sink WSN is suitable for applications that have stringent requirements for data collection and processing, and that need to support a large number of sensors or a large volume of data. A multiple sink WSN is also more reliable and resilient, as it can recover from the loss of a sink or from a partition in the network without disrupting the entire network. It also has less overhead and latency, as the sensors can transmit their data to the nearest sink, which may reduce the traffic and the energy consumption of the network [12]. It also has better scalability, as it can support a larger number of sensors or a larger volume of data without sacrificing the performance or the reliability of the network.

1.3.4 Static-sink and Mobile-sink Wireless sensor Networks

A static sink is a sink that is fixed in a specific location and does not move. It is typically connected to a central node or a cloud server through a wired or wireless connection, and it receives the data transmitted by the sensors through the wireless communication network. A static sink is suitable for applications that require a stable and reliable connection between the sensors and the sink, and that do not require the sink to move.

A mobile sink refers to a sink node that possesses mobility capabilities, enabling it to traverse the coverage area of the sensors. This mobile sink can be transported by a person, an animal, a vehicle, or a drone, and it is designed to collect data from the sensors while in motion. Mobile sinks are particularly advantageous in applications where the sink needs to cover a larger area or be in closer proximity to the sensors, thereby enhancing the quality of signal reception. By leveraging the mobility of the sink, data collection can be more efficient and cater to specific requirements that necessitate dynamic coverage or improved signal quality. Both static and mobile sinks offer unique advantages and face distinct limitations. A static sink provides stability and reliability, unaffected by the mobility and connectivity issues that can arise with a mobile sink. However, a static sink is confined to a fixed location and may struggle to reach or cover all sensors in expansive or intricate environments.

On the other hand, a mobile sink exhibits greater flexibility and adaptability. Its ability to move allows it to cover a wider area and access sensors that may be challenging or inaccessible for a static sink. This mobility is particularly beneficial in applications requiring comprehensive coverage or closer proximity to sensors for optimal signal quality. Overall, the choice between a static or mobile sink depends on the specific requirements of the application. Considerations such as coverage area, accessibility of sensors, stability, and susceptibility to interference need to be evaluated to determine the most suitable sink configuration for a given scenario.

1.3.5 Homogeneous and Heterogeneous Wireless Sensor Networks

In a homogeneous wireless sensor network, all the sensors use the same communication protocol and have similar capabilities, such as range, bandwidth, power consumption, and processing power. However, homogeneous WSNs also have some limitations. They may not be able to support the diverse requirements and constraints of different applications and environments. They may also be less flexible and adaptable, as they may not be able to accommodate the specific needs and preferences of different users and stakeholders. heterogeneous WSNs have more flexibility and adaptability, as they can use a variety of communication protocols and sensors to meet the specific needs and constraints of different applications and environments. They can also leverage the strengths and capabilities of different protocols and sensors to optimize the performance and reliability of the network. However, heterogeneous WSNs also have more complexity and overhead, as they require more coordination and integration between different protocols and sensors, and they may have more interoperability and compatibility issues.

1.3.6 Self-configurable and Non Self-configurable Wireless Sensor Networks

In a wireless sensor network, the term ”self-configurability” refers to the ability of the network to automatically and dynamically configure itself without the need for manual intervention. A self-configurable WSN is a WSN that can adapt and optimize itself based on the changing conditions and requirements of the environment and the application. A non self-configurable WSN is a WSN that requires manual configuration and maintenance, and that does not have the ability to adapt to changing conditions or requirements.

The choice between a self-configurable WSN and a non self-configurable WSN depends on the specific requirements and constraints of the application and the environment. A self-configurable WSN may be more suitable for applications that require a flexible and adaptable network that can adjust its configuration and operation based on the changing conditions and requirements of the environment and the application. A non self-configurable WSN may be more suitable for applications that require a simpler and more stable network which does not need to adapt to changing conditions or requirements.

1.4 Network Architecture

Wireless sensor networks generally follow the OSI architectural model. This paradigm consists of application, transport, network, data link, and physical layers. In addition, there are three components of cross-layer management: power management, mobility management, and task management. These cross-layer components monitor the network and perform various sensor management tasks to ensure that all sensor nodes can communicate efficiently with one another and that network performance is optimized. As discussed earlier, the main purpose of any wireless sensor network is to monitor its surrounding for certain events or phenomenons, collect the required information and transmit this information to the sink node for further processing. There are two major type of architecture which normally used for such kind of communication.

1.4.1 Single-Hop Sensor Network Architecture

A single-hop wireless sensor network is a type of wireless network in which all the sensor nodes in the network are within a single hop of each other, meaning that they are all within range of each other and can communicate directly without the need for intermediate nodes to relay the messages. In a typical one hop sensor network architecture, the sensor nodes are directly connected to the sink node,

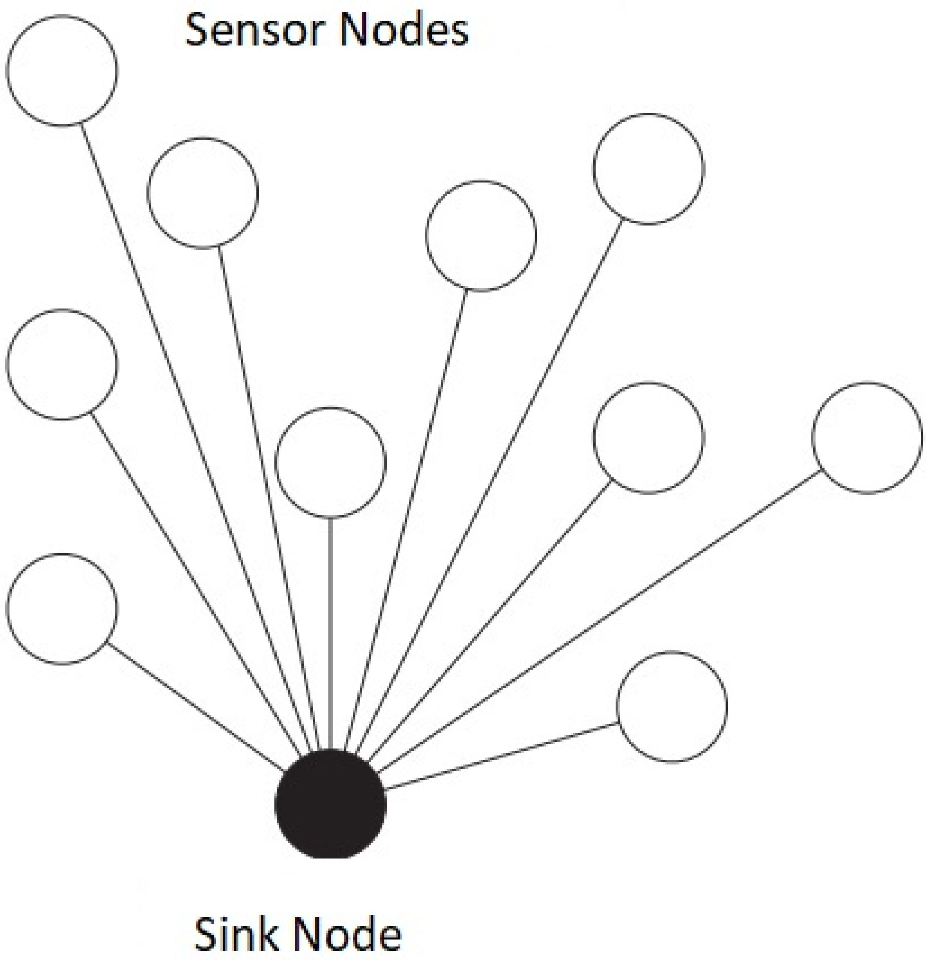

Figure 1.5: Single-Hop Wireless Sensor Network

Figure 1.5. Any sense data over the sensor node is directly communicated to the sink node and vise versa, the sink node sends commands to the sensor node directly to perform certain tasks.

Single-hop wireless sensor networks are typically used in situations where the nodes are all within a relatively small area, such as a building or a small outdoor area. They are often used for applications such as building automation, environmental monitoring, and industrial process control.

The advantages of single-hop wireless sensor networks include low latency, low power consumption, and simplicity of deployment and maintenance. However, they are limited in their coverage area and may not be suitable for larger or more complex environments.

1.4.2 Multi-hop Sensor Network Architecture

This is the most widely used sensor network architecture because of its scalability. All the sensor nodes are not directly connected to the sink node through a single hop, but rather work in coordination to communicate their sense data to the sink node via multiple hops, such that one or more intermediate nodes are utilized to reach out to the sink node. Multi-hop wireless sensor networks are often used in larger or more complex environments where it is not possible or practical to have all the nodes within range of each other. They can be used to cover a wider area, such as an entire campus or a large outdoor area, and can be used for applications such as environmental monitoring, industrial process control, and military operations.

The advantages of multi-hop wireless sensor networks include the ability to cover a larger area and to support a larger number of nodes. However, they may have higher latency and higher power consumption than single hop networks, and they may be more complex to deploy and maintain.

The multi-hop sensor network is generally implemented in two different ways.

1. Flat network architecture.
2. Hierarchical network architecture.

- Flat Network Architecture :

  A flat wireless sensor network is a type of multi-hop wireless sensor network in which all the sensor nodes are considered to be on the same level and can communicate with each other directly, such that each node plays an equal role and has no priority. Each node is autonomous and able to transmit

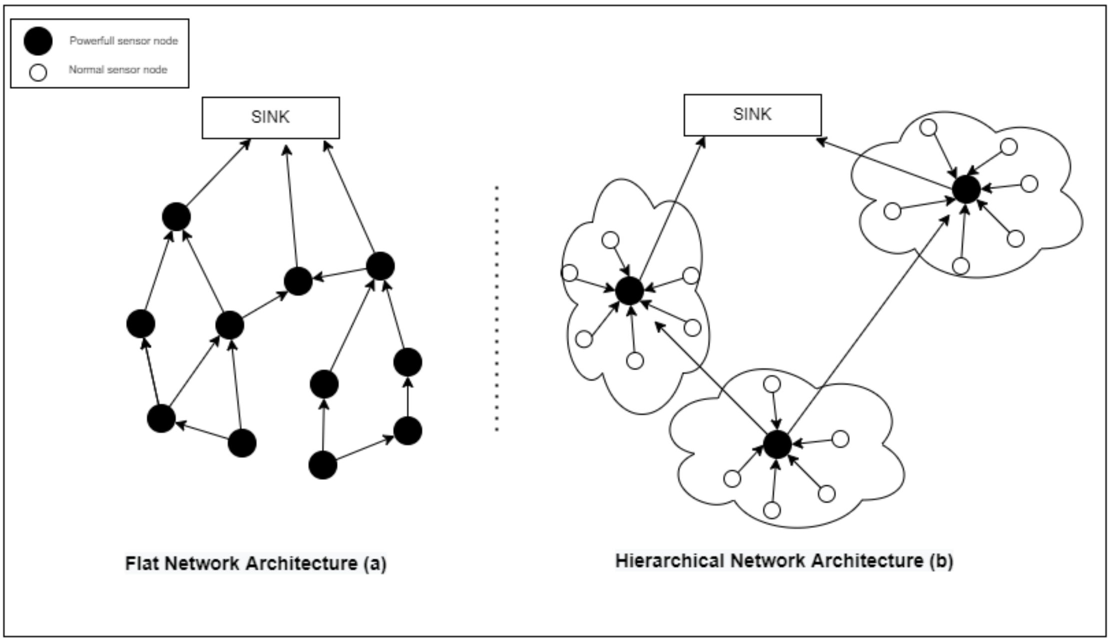

FIGURE 1.6: Multi-Hop Wireless Sensor Networks

information as needed and create its own routing tables. Each node in the network can route any message as needed. In a flat design, each sensor node has the same capabilities. Each sensor node is connected to the sink via a multi-hop path, with peer nodes acting as relays. The network with a standard flat topology is shown in Figure 1.6(a) Flat wireless sensor networks are relatively simple to deploy and maintain, as there is no need to establish a hierarchical structure or designate specific nodes as intermediaries. However, they may have limited scalability and may be prone to congestion as the number of nodes increases. Flat wireless sensor networks are often used in applications where simplicity and low cost are more important than scalability or robustness, such as building automation, environmental monitoring, and industrial process control. They may be suitable for small to medium-sized networks in which all the nodes are within range of each other.

- Hierarchical Network Architecture:

  A hierarchical wireless sensor network is a type of multi-hop wireless sensor network in which the sensor nodes are organized into a tree or other hierarchical structure, with some nodes acting as intermediate relays for other nodes. This is in contrast to a flat wireless sensor network, in which all the nodes are considered to be on the same level and can communicate with each other directly, In a hierarchical architecture, sensor nodes have varying degrees of responsibility depending on their resources. Not all nodes have the same rights and resources. For example, sensor nodes are grouped into clusters across a hierarchical network, with cluster members transmitting data to the cluster head, which in turn transmits it to the sink. A low-energy node is used for sensing and transmitting the sensed information to the cluster head over a short distance, whereas a higher-energy node is used for processing the data from its cluster members and transmitting the processed data to the gateway network, as shown in Figure 1.6(b). Hierarchical wireless sensor networks can improve scalability and reduce congestion compared to flat networks, as they allow some of the nodes to act as intermediaries for other nodes. This can help to distribute the workload more evenly and to reduce the number of direct communication links that are needed. However, hierarchical wireless sensor networks may require more complex deployment and maintenance, as it is necessary to establish the hierarchical structure and designate specific nodes as intermediaries. They may also be less flexible than flat networks, as changes to the network may require changes to the hierarchical structure.

Hierarchical wireless sensor networks are frequently employed in expansive or intricate environments where scalability and robustness are crucial, such as military operations, disaster response scenarios, and industrial process control systems. These hierarchical networks prove particularly suitable for medium to large-sized networks, where not all nodes are within direct range of each other. Consequently, in the case of large-scale networks, hierarchical wireless sensor networks are often preferred due to their ability to efficiently manage communication and data flow across different tiers or levels of the network hierarchy.

1.5 Applications of Wireless Sensor Networks

Wireless sensor nodes are commonly used in many applications that are an important part of the control, management, and in many other applications. Here we are going to discuss some applications of WSN in our daily life.

1.5.1 Industrial Applications

In industrial work, where the heavy work is done through robots. They are controlled by the sensors. The sensors are deployed in the industrial environment to track any problems. Sensors are used in automatic control systems which can sense and report temperature, pressure, output results, and system continuity.

- Auto Mobiles:

  In this age the automobiles are fully controlled by the sensors, VANET (vehicular Adhoc Network) works on the sensor deployed on the automobiles. These sensors intercommunicate with each other and exchange information about the road position, traffics load, about any accident. In Automobile sensors deployment, research is making progress. The driverless car is also a good example of the use of sensors in automobiles.

- Aircraft Control:

  In the Aircraft and Aeroplanes, a sensor networks are used to control and sense critical aircraft systems. Sensors in the aircraft sense the fire, temperature, air pressure, direction, etc. Many sensors are deployed inside and outside of the airplane to collect real-time information. The autopilot aero plan system works with the help of these sensor nodes, which are controlled by remote stations.

- Fire Zone:

  A wireless sensor network can help detect unwanted fires before they can cause damage. For example, fire detection systems are installed in buildings, factories, and natural habitats to provide early warning and take appropriate action to prevent further damage. We can use such sensor networks to monitor places to which humans do not have direct access.

- MIDI Sensor:

  We can also observe the use of wireless sensor networks in the music industry. A special kind of MIDI sensors are attached to the musician's arms and body to form a sensor network. The musical instruments are controlled by the movement of a sensor along with the movement of the musician's body.

- Engineering:

  The use of wireless sensor network in engineering work is very common. Sensor networks are used in Mechanical, civil, electrical, and many other fields of engineering. They are used to calculate the weights, pressure etc. They are also used to monitor any changes. There could be a lot of applications of the sensor network in engineering work.

1.5.2 Environmental Sensing Applications

Wireless sensor networks (WSNs) find significant application in various environmental monitoring scenarios, including habitat monitoring, weather observation, crop and livestock condition monitoring, flood detection, and water management. These applications leverage the capabilities of WSNs to collect real-time data, facilitate decision-making processes, and enhance environmental sustainability and resource management practices.

- Geographical Application:

  Wireless sensor networks are used to locate the geographical location. such as sensor nodes equipped with GPS systems can be utilized to find location , position, dimension of any required object with in the range. Furthermore geographical surveys can be conducted through equipment's comprises of sensor nodes.

- Disaster Management:

  Gyro sensors, Fire sensors, and many other sensors can save from any massive destruction. Gyro sensors can inform about the earth quacks. the sensor network deployed in water can indicate us about any storm.

- Flood and Water Management:

  Wireless Sensor Networks have significant implications for flood and water management. Sensors are used to monitor the water level. These sensor networks measure the flow and level of water in real time. They are extremely beneficial for water control and are typically installed on bridges and barrages.

- Weather Monitoring:

  Weather forecasting is also an application of Wireless Sensor Networks. Sensors measure the air pressure, air moisture, temperature, and intensity of gases. They can measure any change in the environment.

- Habitat Monitoring: Habitat monitoring is an important application of wireless sensor networks used to monitor the habits and natural environment of animals and birds. The goal is to collect accurate data on the natural habitats of animals with minimal human interference. The sensor nodes are deployed in the forest or on the animals to observe the original natural wildlife. It provides researchers with a very good opportunity to observe animal behavior in the wild.

1.5.3 Home and Office Applications

Wireless sensor networks have a wide range of potential applications in both home and office settings, including:

- Home security:

  WSNs can be used to monitor and protect homes from potential threats, such as burglaries, fires, and floods. Sensors can be placed around the home to detect movements, sounds, or other signs of an intruder, and can alert the homeowner or authorities if necessary.

- Home automation:

  WSNs can be used to automate various systems in homes, such as lighting, heating, ventilation, and appliances. This can help to improve energy efficiency, comfort, and convenience.

- Office security:

  WSNs can be used to monitor and protect offices from potential threats, such as burglaries, fires, and flooding. Sensors can be placed around the office to detect movements, sounds, or other signs of an intruder, and can alert the office staff or authorities if necessary.

- Office automation:

  WSNs can be used to automate various systems in offices, such as lighting, heating, ventilation, and appliances. This can help to improve energy efficiency, comfort, and convenience.

- Meeting room scheduling:

  WSNs can be used to manage the scheduling of meeting rooms in office buildings. Sensors can be placed in the meeting rooms to detect when they are in use, and the availability of the rooms can be displayed on a central dashboard or on a mobile app.

  Overall, WSNs have the potential to improve the security, efficiency, and convenience of both home and office settings.

1.5.4 Security Applications

Wireless Sensor Networks (WSNs) are extensively utilized in the domain of security applications, playing a pivotal role in monitoring and control systems. These sensor networks find application in a diverse range of security systems, enabling effective surveillance and control.

- Missile Technology:

  Wireless sensor networks are employed in missile technology. Sensor nodes built within missiles collaborate as a Wireless Sensor Network to find the target, determine the missile's orientation relative to the target, and determine the missile's speed. Sensor networks play a key role in the evolution of missile technology.

- Military Application:

  Numerous military applications make use of wireless sensor networks. They are used to locate the moment of troops, metal detector sensor which can detect any metallic body even underground. This is very helpful in the detection of hidden weapons. They give and take commands to military troops. Automatic weapons are dependent on sensors and they can perform an important role in any terrorist attack.

- Surveillance Application:

  Wireless sensors are used for surveillance purposes. A video sensor is very small in size and the camera is embedded in sensors that can give the footage of its coverage area. Video sensors can be used in critical situations for valuable information gathering.

1.5.5 Health Care Applications

Sensors are now widely used in medical applications. WSN is used to monitor and collect data from the patient remotely. They are capable of easily measuring blood pressure, temperature, and blood sugar levels, among other things. Some of the potential applications in healthcare are:

- Patient monitoring:

  WSNs are used to monitor the health and well-being of patients, such as by tracking vital signs, detecting falls, and alerting caregivers. This can help to improve patient care and reduce the burden on the healthcare system.

- Telemedicine:

  WSNs are used to facilitate remote consultations between patients and healthcare providers, such as through video conferencing or remote monitoring. This can improve access to healthcare for patients in remote or under-served areas.

- Clinical trials:

  WSNs are to collect data from patients participating in clinical trials, such as by monitoring their symptoms or tracking their compliance with treatment protocols. This can help to improve the efficiency and accuracy of clinical trials.

- Hospital operations:

  WSNs are used to improve the efficiency and safety of hospital operations, such as by monitoring equipment, tracking the movements of staff and patients, and automating various processes.

- Public health:

  WSNs are used to monitor and track the spread of infectious diseases, such as by collecting data on the movements and behaviors of individuals. This can help to inform public health strategies and interventions.

1.5.6 Agriculture Applications

- land monitoring:

  Sensors are used in Agriculture and they are also used to check the field's level. Small sensors are used to monitor the fields. A sensor can report about plant health, moisture, and fertilizer needs. In advance, agriculture equipment's multiple sensors are deployed.

- Greenhouse Monitoring: Greenhouses are botanical gardens, where plants are grown in a controlled climate. Wireless sensor networks are widely used to maintain an operational greenhouse, where different kind of specialized sensors are used to monitor temperature, humidity, moisture detection, automatic watering of plants.

1.6 Highlighted Issues in WSNs

Wireless sensor networks consist of a large number of sensor nodes with limited computational power and battery life. Since the lifetime of each sensor node is unknown, the network must be autonomously constructed, as manual configuration of the sensor network is impractical for many applications. In addition, the sensor network may consist of a number of different types of nodes, highlighting the need to consider heterogeneity. The number of sensor nodes in a network does not remain constant during its lifetime; it may change when new sensor nodes are added or when nodes are removed due to death. The following are the most important aspects to consider when designing a sensor network.

1.6.1 Fault Tolerance

Fault tolerance is a critical aspect of system design, particularly in the context of Wireless Sensor Networks (WSNs). WSNs are often deployed in demanding and unpredictable environments, making them susceptible to node failures and communication disruptions caused by natural events or external factors. To ensure uninterrupted operation and data integrity, it is essential to incorporate fault management strategies into the design of a robust wireless sensor network.

Implementing fault management strategies involves proactive measures to identify potential failure points and develop mechanisms for fault detection, isolation, and recovery. This may include redundancy in sensor nodes, fault detection algorithms, fault-tolerant routing protocols, and data replication techniques. By taking a proactive approach to address potential failures, the network can maintain continuous operation, minimize data loss, and mitigate the impact of component failures.

1.6.2 Scalability

In a wireless sensor network, we generally deal with two types of scalability. The first is hardware scalability, which refers to the number of deployed nodes, their sensitivity, communication range, radio bandwidth, and energy consumption. The second is software scalability, which includes scalable routing protocols, data transmission capacities, processing capacities, and sensor node management. There is a tradeoff between the increasing number of nodes deployed and the increasing complexity of management.

1.6.3 Connectivity

As mentioned earlier, wireless sensor networks are used in a variety of industries for monitoring and surveillance, such as medical, security, and risk assessment. All of these applications require a means of transmitting accurate data in real time in order to function properly. In a multi-hop wireless sensor network, nodes are interconnected via a wireless medium, which introduces a number of inherent problems, such as node failure, fading, high error rate, scattering, and other classic problems with wireless media; Which affects the connectivity and wireless sensor network unable to operate according to the application requirements.

1.6.4 Energy

Energy consumption is a critical aspect to address in wireless sensor networks, as it directly impacts the network's overall lifetime, which is contingent on the longevity of individual sensor nodes. The major contributors to energy consumption in WSNs are the sensing, analysis, and transmission of data to a central repository. Consequently, effective energy management encompasses the development of algorithms, protocols, and deployment strategies aimed at minimizing the energy requirements in sensor nodes and ensuring the sustained operation of the network. By optimizing energy usage, WSNs can achieve prolonged network lifetimes and maximize the utility of available resources.

1.6.5 Topology/Network Model

The topology of a network refers to the arrangement of sensor nodes in a given area relative to each other. The architecture of the sensor network can be stable for static nodes or flexible for mobile sensor nodes. Localization and relative position of nodes are critical to coverage and connectivity in a wireless sensor network. A changing topology can occasionally lead to network holes, which induce disconnection and adversely affect upper layer applications.

1.6.6 Security

Another important feature of WSN is security. Data sensitive applications, such as medical monitoring, require protection of data from forgery and eavesdropping. Being a resource-constrained network, wireless sensor networks require more sophisticated, specialized security techniques. Protocols that incorporate cost-effective encryption algorithms are critical.

1.6.7 Deployment

Wireless sensor networks often involve the deployment of a substantial number of nodes in a targeted area. The seamless collaboration and coordination among these nodes are paramount to the network's overall efficacy. The relative positions and locations of the sensor nodes play a pivotal role in determining the network's performance and functionality. As a result, meticulous planning and thoughtful deployment strategies are essential elements in the design of wireless sensor network architectures. By strategically considering the placement of sensor nodes, the network can achieve optimal coverage, efficient data collection, and robust communication, thereby enhancing its overall effectiveness and reliability.

1.7 Research Objectives

The main motivation of this research is to examine the impact of node failures on the overall connectivity of wireless sensor networks. These node failures can result in network partitioning, causing the network to become fragmented into multiple network partitions, and each of these isolated parts can lose connectivity to the main part of the network where the sink resides. Therefore, identifying and reconnecting these isolated partitions is an important aspect of wireless sensor network functionality.

Our goal is to analyze various partition identification and re-connectivity techniques available in the literature in order to develop a reliable partition identification and re-connectivity mechanism, which can perform partition recovery with limited use of additional resources in a timely manner with minimal delay. Thus, to minimize data loss, provide fault tolerance in order to increase network lifetime.

In this context, we aim to investigate the innovative use of collaborative beamforming in homogeneous wireless sensor networks, sensor node energy visualization alongside traditional solutions, in order to provide a viable solution to the network partitioning problem.

1.8 Connectivity in WSN

Connectivity in is one of the most important design constraint/issue of WSN. It has a significant impact on the QoS aspects of the network [13]. The main task of a wireless sensor network is to fully monitor the area of interest with a minimum number of nodes. Therefore, optimized coverage is required. Moreover, the information collected by the sensor nodes must need to be forwarded to the sink, which is only possible if a valid path to the sink exists. Since the sensor nodes are very vulnerable and are mostly deployed in remote areas, there is a high probability that some of them will be damaged and create coverage holes. In the WSN network, hop-by-hop communication takes place, where many nodes cooperate to forward information to the sink. Therefore, any damage to a node that is on the path to the sink can cause a disconnection. One of the main causes of node failure in a WSN is energy depletion. Energy consumption plays a significant role in maintaining connectivity and maximizing the lifetime of the network. The researcher addressed coverage and connectivity problems through efficient deployment, putting sensors into sleep mode to reduce energy consumption, and adjusting sensing range to conserve energy. Below are some of the main causes of network connectivity interruption.

1.8.1 Causes of Network Dis-connectivity

Some of the major causes of the network dis-connectivity are as following.

- Random deployment result in uneven density over the field of interest.
- Node failure due to energy depletion.
- Bottleneck paths within the network.
- Sensor hardware issues.
- Overheating, other environmental causes which result in sensor damage.
- Terrain variation, blockages within the communication path.
- Dynamic topology in a mobile wireless sensor network.
- Routing mechanism based on shortest path matrix may result in a bottleneck.

1.8.2 Void Regions

A void region is defined as an area within the region of interest that is not monitored by sensor nodes [14]. This type of void region is typically the result of inefficient deployment. The main causes of sensor node loss are the effects of deployment, hardware failures, or energy depletion. Each of these phenomenons has an impact on the overall performance of the network.

Void regions can have a negative impact on the performance and reliability of a wireless sensor network, as they can prevent the nodes in the void region from communicating with the rest of the network. This can lead to gaps in the data collected by the network, which can affect the accuracy and usefulness of the data. These network holes can also affect routing if they are located along a critical path [15]. Moreover, these vacant regions in the network may result in network partitioning, simply dividing the network into numerous small fragments.

There are several ways to address the void region in a wireless sensor network, including additional node deployment, increasing the range or power of the nodes and using mobile nodes. Furthermore, the problem of network hole/void region are usually treated in three different ways, I.e. geometric computations, statistical computations, and algebraic topological computations [16]. In geometric computational approaches, Voronoi diagrams are used. Similarly, in [17] the authors presented a cluster-based hole detection by utilizing hybrid deep learning algorithms.

1.8.3 Significance of Network Connectivity

Network connectivity improves healthcare operations. With the advancement of IoT devices, connectivity is becoming increasingly important for today's medical gadgets. A strong wireless network connection is critical for healthcare personnel and patients to provide consistent and appropriate care throughout the facility. [18][19]. Healthcare personnel require consistent wireless connectivity across all devices, all the time, whether they are transmitting patient records, contact information, test results, or doing other wireless-dependent. Security applications rely on real-time monitoring data to detect security breaches. Any dark spot can lead to a disaster, so connectivity and network coverage are paramount. Industrial applications also require reliable connectivity for real-time monitoring of critical machinery and industrial waste.

1.9 Network Partitioning

Network partitioning is a term that refers to the division of wireless sensor networks into numerous segments. This results in connectivity issues. There is a possibility that some of these isolated segments will not find a valid path to the sink, so they cannot send important sensor data. The occurrence of void regions in the network due to node failures is a major cause of network partitioning, as these void regions may occur at critical points along the network's path to the sink.

1.9.1 Effects of Network Partitioning

Network partitioning has several negative consequences, including connectivity loss, loss of critical data, reduced coverage, increased routing convergence costs, energy depletion due to packet loss, excessive control packet overhead, and a reduction in overall network lifetime.

1.9.1.1 Connectivity Loss

A wireless sensor network is all about monitoring its area of interest, gathering information, and communicating it to a central repository for further processing and decision-making. Therefore, coverage and connectivity are the most important aspects in the development of a wireless sensor network. The loss of connectivity due to the failure of multiple nodes makes it impossible to transmit sensor data to a central repository. Network partitioning is one of the major culprits for this undesirable behavior, it leads to reduced network coverage, limited data availability, and compromised system performance, emphasizing the need for effective strategies to address and mitigate this issue.

1.9.1.2 Coverage Loss

As mentioned earlier, one of the main objectives of any wireless sensor network is to monitor its environment and collect relevant data with respect to the application requirements. Therefore, once the network is partitioned, it is more likely that coverage of a particular area of interest will be lost. Numerous void regions established within the network result in isolated, uncovered fragments within the area of interest.

1.9.1.3 Routing Overhead

Routing protocols are responsible for navigating data packets within a network. They rely primarily on the topology of the wireless sensor network to find the most appropriate path for these data packets to reach their desired destination. Once the sensor detects and collects the required data, it is the responsibility of a routing protocol to find the most appropriate path to the central repository and then navigate through the network. Therefore, any change in the topological structure of a network due to network partitioning will trigger the alternate path finding mechanism, which may cause control packet overhead.

1.9.1.4 Critical Data Loss

When a network becomes partitioned, the isolated fragments become disconnected from the central repository, impeding their ability to transmit sensed data. This can have severe consequences, particularly for data-intensive applications like medical and surveillance systems, where the loss of critical data can have detrimental effects. Furthermore, the resulting packet loss, caused by unavailable routing paths, exacerbates the energy depletion problem in sensor nodes, further compromising their operational lifespan and overall network performance. Mitigating network partitioning is crucial to ensure uninterrupted data transmission, maintain data integrity, and alleviate the strain on sensor node energy resources.

1.9.1.5 Network Lifetime

The lifetime of a network depends on the efficient use of network resources to accomplish the required task as efficiently as possible. Therefore, resource utilization is critical to the lifetime of the network. Once a partition occurs in the network, error detection and recovery mechanisms such as routing convergence, packet rescheduling, and control packet overhead often occur, leading to increased resource utilization and ultimately contributing to accelerated energy depletion of sensor nodes.

Since a wireless sensor node's transceiver consumes a lot of energy when sending and receiving data packets; therefore, packets that do not reach their intended destination are vulnerable to sensor nodes with limited energy. The overall life expectancy of the network depends on the number of active, functioning sensor nodes. Each prolonged node failure eventually reduces the efficiency and lifetime of the network.

1.9.1.6 Restoration Cost

Sensor network restoration is an expensive task, specifically while dealing with a hostile environment. To recover from a network partitioning, we need to restore coverage over the faulty void region. This is usually done by deploying new nodes or rearranging the existing sensor nodes so that the partitioned network can be reconnected with minimal network rearrangement and coverage loss. Real-time data centric applications suffers more in such situation.

In light of these challenges, it becomes essential for an efficient wireless network to possess the capability to either prevent network partitioning altogether or effectively overcome it when it occurs. Such a network must exhibit robustness, maintaining continuous connectivity even in the face of potential partitioning events. By prioritizing connectivity and implementing robust strategies for partition recovery, wireless sensor networks can ensure uninterrupted data flow and optimal performance, enhancing the reliability and effectiveness of real-time applications.

1.10 Beamforming

The connectivity among the wireless sensor nodes depends on the communication range of sensor nodes, i.e. how far a sensor node can transmit its signal to be detected by the neighbor node in order to relay it to the sink. Thus, a network can be assumed highly connected if the sensor nodes are densely deployed or the sensor nodes have a high communication range. Beamforming is a special type of radiation pattern of an antenna that can be achieved by combining multiple radiating elements. When we feed the same signal into multiple radiating elements of an antenna, it results in a strong signal in a particular direction, rather than having a spread signal in all directions from a broadcasting antenna. In this way, a signal can be steered over a long distance [20]. This technique was first used to improve sonars in World War 2, and since then it has been used in all types of wireless networks and communications. Nowadays, it is also used in wireless sensor networks and emerging 5G MIMO networks. There are two main types of beamforming as follows.

- Localized Beamforming
- Distributed Cooperative Beamforming

1.10.1 Localized Beamforming

The localized beamforming refers to the formation of a directed energy beam using a group of radiating elements/antennas. Each antenna that generates a signal has a specific phase that depends on the distance between them. If we somehow tweak the phase of the signal by introducing certain delays and make them all identical, they will perform constructive interference and the resultant waveform will be a powerful radiation pattern called a beam [21]. Beamforming is based on the principle of reciprocity, i.e., we can send a high-power beam in the desired direction to improve reliability while carefully listening in the direction of the transmitter to receive an amplified signal due to constructive interference and suppress the noise of the transmitter, as shown in Figure 1.7.

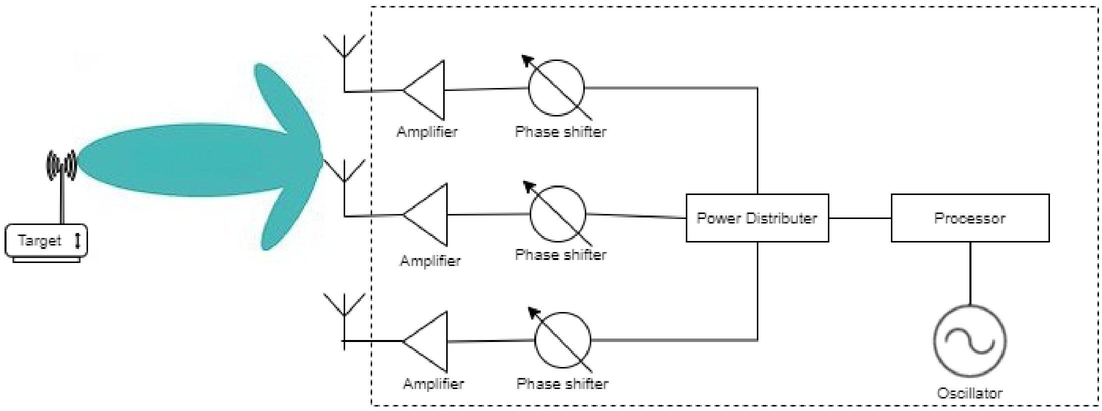

FIGURE 1.7: Localized Beamforming

1.10.2 Distributed Cooperative Beamforming

Distributed collaborative beamforming is a special implementation of beamforming in which multiple randomly placed nodes collaborate with each other to form a virtual antenna array that performs the beamforming [22–24]. The orientation of the antenna elements in this beamforming system is generally random because the location of the collaborating nodes, which act as virtual antenna array elements, is not localized. Each node is a self-contained entity. Therefore, each participating sensor node antenna has its own processor and oscillator. As a result, each antenna element initially has its own unique phase, frequency, and time reference.To perform beamforming, these randomly distributed nodes must synchronize their frequency, phase, and time accordingly, as shown in Figure 1.8. In terms of increased data rate and transmission range, a virtual distributed antenna array formed by a cluster of sensors has some significant advantages. According to Shannon's formula, the beam created by an antenna array contains more radiated energy focused at a certain direction, which improves the signal-to-noise ratio at the destination, resulting in increased channel data rate. This virtual antenna array also makes use of the power of multiple input-output systems (MIMO), which results in further data rate gain. The antenna gain of an array of elements can

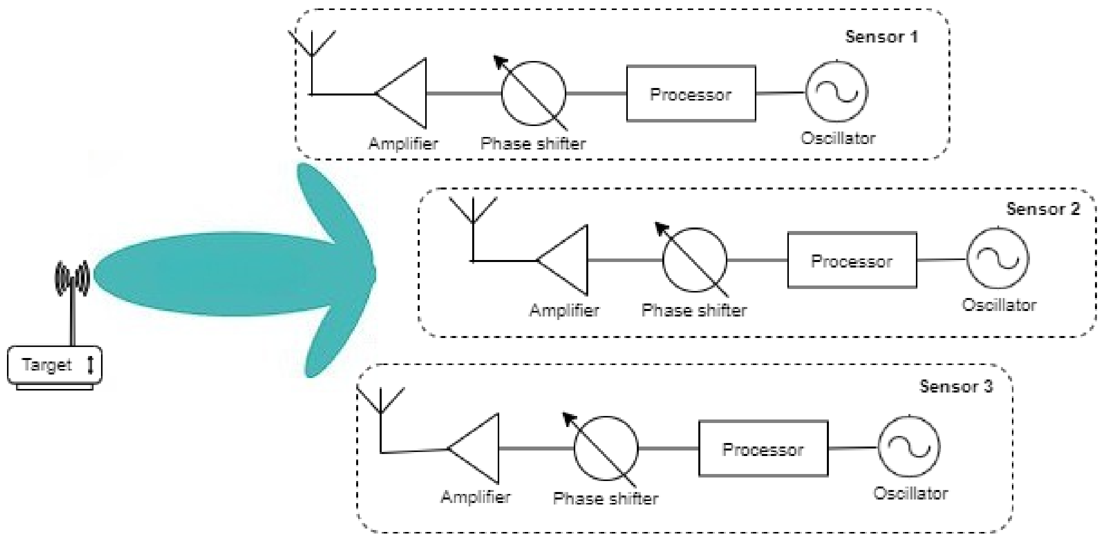

FIGURE 1.8: Distributed Collaborative Beamforming

be several orders of magnitude greater than that of a single element. Thus, it subsequently increases the communication range of the sensors by maintaining an acceptable, fixed signal-to-noise ratio at the destination. The directional beam also performs spatial filtering, which reduces signal interference from adjacent signals. In addition, the ability to select the number of sensor nodes as array elements to perform cooperative beamforming allows us to customize the antenna gain/communication range enhancement based on our requirements. This control over antenna elements enables us to performed energy efficient beamforming. Collaborative beamforming using a cluster of randomly distributed sensor nodes is distinct from conventional beamforming with respect antenna array elements, as illustrated in Table 1.2. There are three primary considerations when designing an antenna array system for beamforming. 1. Antenna element count; 2. Antenna element position. 3: synchronization of the array elements' phases. In conventional beamforming, the number and position of the antenna elements are fixed and known in advance; a dedicated central DSP processor handles synchronization and beamforming coordination in order to construct the ideal beam in the desired direction. On the other hand, identifying the sensor nodes that will participate in collaborative beamforming is a distributed by nature, and their positions are random. Additionally, the number of participating nodes varies, and we will need to deal with signal synchronization. Therefore, these variations make distributed

Table 1.2: Virtual Antenna Array VS Localized Antenna Array

Description	Virtual Antenna Array	Localized Antenna Array
Array establishment	Distributed	Localized
Beamforming Coordination	Distributed	Localized
Number of Elements	Random	Fixed
Element Position	Random	Fixed
Element Position Error	High Probability	low Probability

beamforming with sensor clusters is a more challenging task as compared to traditional arrays, but the benefits of a higher signal-to-noise ratio and range extension gains from beamforming are significant.

1.11 Wireless Network Visualization

A visualization of a wireless network can be utilized to detect and prevent network anomalies such as network partitioning, void regions caused by coverage holes, security wormholes etc. A visualization application is typically used to determine and manage the state of a wireless network. The visualization software receives data from wireless devices in a wireless network, plot each wireless device in 2D space, and creates a visualization that displays the data from the wireless devices and the network. To manage the network, wireless devices are selected on the visualization application screen, and control activities are performed. Visualization of connections, traffic, and information about wireless devices are the three most typical visualization functions in a visualization application. Most visualization programs implement only a subset of these functions, depending on their purpose. A general depiction of a typical network visualization can be seen in Figure 1.9 Information visualization is a technique that has been utilized for a long time to graphically depict information in a clear and effective manner [25]. In order to extract information, the fundamental objective of visualization is to generate

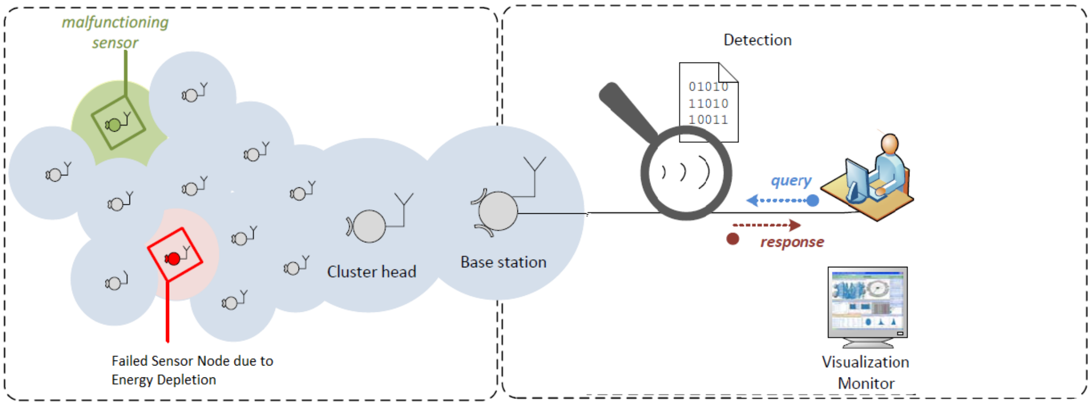

FIGURE 1.9: Wireless Sensor Network Visualization

interactive visual representations of data that leverage human perceptual and cognitive problem-solving abilities. This is why a rising number of academics support the role of visualization as a new data mining tool, often known as visual data mining [26] capable of resolving complicated issues. Information visualization has been utilized in a variety of industries, such as, for network data visualization. Network traffic visualization is likely one of the first steps to take when it comes to comprehending, evaluating, and locating pertinent information among a network's massive volumes of data. Numerous visualization strategies (scatter plots, color maps, various types of graphs, etc.) have been offered by researchers to assist network analysts with this task [27]. Several network visualization tools for graphically monitoring real-world or simulated sensor network deployments have been presented in the field of WSNs. Existing tools, such as the Surge [28], MoteView , Octopus, Spyglass and the Sensor Network Analyzer (for ZigBee-based WSNs), display network activity between sensor nodes and provide users with real-time information about the network topology and the collected sensory data to enable real-time debugging of the deployed sensor network. In addition, these tools include a variety of visualization features that make it simple to discover packets of interest and monitor the network's health and performance. Visualization can also be useful for studying and comprehending network simulations. Several network simulators suited for WSNs (TOSSIM,OPNET,NS-2, and QualNet) have begun to use visual representations of the network topology layout and packet level animation. Typically, these simulators generate enormous files comprising network traffic statistics, sensor readings, radio links, and topology layout information, which simplifies the user's network research process.

1.12 Thesis Organization

The rest of the thesis report is organized as follows: in Chapter 2, the state-of-the-art related to WSN is discussed, including WSN deployment, network partitioning, beam formation, and network re-connectivity techniques. Some open research issues are also discussed in this chapter. The focus is given on the different techniques proposed to identify network partitioning and mechanisms to reconnect isolated partitions to the main part of the network. Chapter 3 focuses on the identification of partitions in a randomly deployed wireless sensor network. In Chapter 4, a collaborative beamforming mechanism for partition detection and establishing connectivity is proposed. In Chapter 5, a method for visualizing sensor node energy information is proposed to facilitate the network manager finding vulnerable sensor nodes and replacing them with additional relay nodes to ensure connectivity and avoid network partitioning. In the end, Chapter 6 concludes the thesis and discusses future directions for the extension of the proposed theory.

Chapter 2

Literature Survey, Gap Analysis and Problem Formulation

2.1 Related Work

Wireless sensor networks face a variety of challenges in order to achieve the objectives for which they are designed. This is particularly true when a single sensor node or a collection of sensor nodes fail, due to which a part of the network is disconnected. The aforementioned subject has been widely investigated in recent years and is of the utmost importance. When sensor nodes are deployed in a harsh and hostile environment, such as a battlefield, enemy territory, or an area inaccessible to humans, such as deep waterways or volcanic mountains, this problem becomes much more significant and difficult to control. Numerous approaches have been developed in response to this issue. Generally, the researchers handled the topic in two stages.

- Identifying void regions created by node failure, in which the sensor network loses coverage and acts as a network hole, which is believed to be a main source of Network disconnection/Network partitioning.

- Determining the optimal method for resuming coverage over that vacant area in order to prevent network partitioning and eventually restore connectivity.

In general, sensor nodes are randomly dispersed in uncontrolled regions by airborne deployment. This deployment method may result in an uneven and unsupervised placement of sensor nodes inside the Area of interest. Maintaining sensor network connection and coverage is critical for achieving the WSN's objectives. We arranged our review of the literature into three sections, namely:

1: Network Partitioning 2: Conventional Re-connectivity Solutions 3: Collaborative Beamforming in Wireless Sensor Networks

2.1.1 Network Partitioning

Network connectivity is one of the main design goals of wireless sensor networks. The sole purpose of the sensor nodes deployed in the area of interest is to sense the surrounding environment according to the application requirements and communicate the sensed data to some central repository for further processing and decision-making. Optimized connectivity plays an important role in network efficiency and lifetime. The loss of connectivity caused by the failure of the sensor nodes on a large-scale has a significant impact on system performance, as it causes additional overhead or, in extreme cases, leads to the complete collapse of the active network.

As discussed earlier in chapter 1, the wireless sensor network generally follows the hop-by-hop communication pattern, i.e., all the nodes work in collaboration with each other to relay information to the sink. Therefore, large-scale nodes' failure leads to connectivity issues [29]. The communication/routing holes created in the network cause the active network to split into many isolated subnetworks/partitions [30]. The network partitioning was first highlighted as a potential threat to any deployed wireless sensor network in [31] by Chong *et al* in 2003, while discussing the evolution of WSN. The authors realized that the main part of the network always needs a connection with the disconnected clusters of the network, which are deprived of communication due to the unavailability of an active path to the sink located in the main part of the network. They suggested that appropriate schemes should be developed to deal with the network partitioning problem. In [32], the authors presented the idea of self-configuring network topologies for a wireless sensor network, focusing on the importance of network partitioning. In one of the earlier works on partition detection, Kleingbeg *et al.* in [33] proposed the idea of deploying special sensor nodes, called agents, at different locations in the network. These agents exchange control messages with each other to check the network connectivity and liveness. If the number of control messages falls below a certain threshold, it is assumed that a cut in the network has occurred. Similarly, in [34], the authors extended the work of [33] by proposing sentinel nodes, a counterpart to agents, to detect network partitions. They consider only linear cuts. The cut identification was only possible at the sink node. It was a centralized technique that required complete information about the network topology. Ritter *et al.* In [35], proposed a reactive partition detection algorithm. They used border nodes to identify a partition. In this technique, an alive message is propagated into the network and awaited for its response. The nodes on the boundaries that do not receive alive messages within a certain threshold time are assumed to be partitioned. In [36][37], the authors propose a distributed state-based algorithm for detecting cuts in the network. In this method, each node in the network contains state information about its neighbors. A positive state value of the node indicates that the node is connected to the sink through its neighbor nodes, while a state value of zero means that the connection to the sink node is broken. Ranga *et al* [38] investigate the identification and recovery of partitioned WSANs. The approach proposed, Distributed Partition Detection and Recovery Using Unmanned Aerial Vehicles, is divided into three phases: initialization, operation and detection, and recovery. By performing distributed failure detection and sending only crucial update messages to the sink node, the strategy is aimed to minimize computation and message overhead on the sink node. The first phase identified cut vertex nodes via the first hop message exchange and sent this list to the sink node. When a cut vertex node fails, the sink assumes a partition has occurred and initiates the recovery procedure. Korien and Bayoumi researched the detection of holes in wireless sensor networks, which is the main reason for network partitioning [39]. They created a technique for finding wireless sensor holes called Wireless sensor Hole Detection (WHD). They divided the region of interest into numerous grids, chose a few reference sensor nodes within each grid, then used triangulation to find network holes by connecting the selected sensor nodes with cell edge points. The vacant zone within each grid that is not covered by any sensor node is referred to as a coverage hole in a network. A proactive partition detecting mechanism is proposed in [40]. The hop count-based cut detection algorithm calculates the minimum hop count and link cost of each node to the sink node using the information embedded in the Hello packet. In [41], the author proposed a sleep scheduling based energy conservation technique to prevent network partitioning. To conserve energy and prolong network lifetime, this technique uses a tree mechanism. The high energy nodes are placed as root branch nodes with actively participate in communication, while the low energy nodes are set as leaf nodes and can put to sleep to conserve energy. The Coverage-aware and Connectivity Constrained Actor Positioning (C2AP) algorithm is introduced in [42]. The authors of this approach talked about nodes after they were deployed. Due to the lack of a definite distance being specified and the potential loss of connectivity, isolation of one or more nodes may happen with this strategy. When a network node fails, this method offers no recovery strategy. In [43] the authers proposed an energy-aware algorithm for coverage and connectivity of sensor nodes to by deploying multiple set of redundant sensor nodes to maximize the coverage. The algorithm addresses the challenge of ensuring connectivity and coverage maximization while considering energy constraints in the network. It relies on multiple cover sets of sensor nodes to maintain connectivity among the sensor nodes.

Furthermore, few of the researchers also taken into consideration the environmental factors to achieve realistic scenarios for sensor node deployment and minimize the coverage problems. Akbarzadeh *et al* [44] first time introduced the importance of environmental factors in wireless sensor network deployment optimizations.They used evolutionary algorithms to evaluate the optimized coverage of the sensor nodes by taking into account the environmental effects. The proposed scheme only relies on sensor vision, i.e. line of sight coverage. In [45] the authors proposed a probabilistic sensing model. By integrating the terrain information, they tried to achieve the maximum coverage probability. Functions for sensing range and sensing angle were derived. The model is then tested with different optimization schemes. In [46], a voronoi based optimization technique has been proposed for sensor deployment by incorporating real world environment. A computational geometry based solution was provided, The authors used Voronoi diagrams and Delaunay triangulation to determine the coverage range of a sensor. Similarly, in [47] the impact of spatial data quality has been investigated by analyzing spatial 3D city models.The authors discussed the accuracy and validity issues related to collection and construction of special data. Furthermore, the impact of 3D data quality on the concept of veiwshed and line of sight has been discussed. In [48] the authors used genetic algorithm to solve the complexities generated through incorporating environmental factors. They proposed a parsing crossover method to cope with terrain irregularities to optimize sensor deployment.

2.1.2 Traditional Re-connectivity Solutions

Partition reconnection deals with the techniques for reconnecting the detected disconnected partitions containing alive sensor nodes to the main part of the network, still connected to the sink node. Thus, network connectivity is one of the major design issue of wireless sensor network.

There are certain strategies to heal the network partitioning through node redeployment and node relocation. We can categorize partition recovery techniques into three major categories.

- Redeployment based solutions
- Mobility based solutions
- Adjustable Transmit Power based solutions

2.1.2.1 Redeployment Based Solutions

Redeployment-based partition recovery solutions aim to address the issue of network partitions in wireless sensor networks (WSNs) by strategically repositioning or redeploying sensor nodes to restore network connectivity and communication. These solutions involve algorithms and mechanisms to detect network partitions, identify critical nodes for redeployment, and determine optimal locations for their relocation. Following are some notable redeployment based solutions. In [49] Ranga *et al.* proposes UAV-BITS as an innovative approach for detecting and recovering network partitions in damaged sensor networks. The solution leverages multiple UAVs and employs Connected Component Sets (CCS) created iteratively to identify partitions, while also establishing fault-tolerant paths between partitions among the closest border nodes. They two-vertex disjoint paths between the network partitions. They aim to provide a reliable connectivity through optimized number of additional nodes with limited UAV mobility. Wang *et al.* [50] presents a redeployment mechanism by finding out the most vulnerable cluster sets where redeployment will contribute to prolonging the network lifetime and optimize network connectivity. In their proposed technique, they take into account two new parameters which were not directly used in the previous techniques, i.e. probability of event occurrence in the cluster and the routing mechanism to be used to propagate those events to sink. They proposed heuristic algorithms and use expectation based prediction approach to solve the redeployment problem. In [51], the RSM(Response Surface Methodology) with desirability functions has been evaluated for optimization of WSN deployments. The authors used case studies for evaluating their approach. In another work, Ko *et al.* [52] focus on macroscopic aspect of network and argues that it is better to exhaust individual node to achieve the better overall connectivity of the network. The authors propose two algorithms, one for optimum density distribution and other for convolution. Bhatt *et al.* [53] used the Markov process to find out the probability of node redeployment and applied stochastic processes for recovering the holes in the network. A face topology-based WSN coverage and connectivity restoration strategy is presented in [54]. In the event that a crucial node fails, they use extra redundant nodes to ensure network connectivity. This method requires more redundant nodes in exchange for fault tolerance. In [55, 56], the authors address the network partitioning problem through additional relay node deployment. In their research, the focus is on finding the optimal number of relay nodes and identifying the suitable location for these additional relay nodes to be placed. similarly, in [57], the partitioning problem is also addressed through efficient redeployment of additional relay nodes using concentric format points. In [58], two fault-tolerant algorithms were proposed to restore connectivity through additional relay node deployment. The proposed algorithms utilize minimum convex hull to obtain several other nodes to be used along with their respective deployment position to achieve network connectivity. A partition recovery scheme using centroids is proposed in [59]. Centroids are used utilized to design a route for relay node deployment. A network holes detection and healing solution is proposed in [60] through powerful and costly relay nodes. In [61], the authors present an anomaly detection system based upon isolation principle. To maintain connectivity and ensure data reliability, they consider the spatial correlation among neighboring sensors to detect node/s isolation. [62] Proposed a similar iterative algorithm to solve the connectivity problem. They used combinational optimization in their proposed approach.

In the above-discussed approaches, static node redeployment was the focus of research. In this regard, the main focus of the researchers was to identify three basic parameters. 1) when to redeploy the additional nodes. 2) the number of required sensor nodes. 3) location calculation for the redeployment of other nodes to overcome the network partitioning.

The major disadvantage of these approaches is the requirement for additional resources to address network partitioning poses a significant drawback to these approaches. The allocation of extra resources not only increases the cost and complexity of the system, but also introduces potential delays in restoring network connectivity. As a result, the time-consuming nature of node redeployment renders these approaches ill-suited for time-critical data applications where prompt and uninterrupted data transmission is paramount.

2.1.2.2 Mobility Based Solutions

Contrary to the redeployment based solution, some researches investigated node mobility for network restoration. To improve network performance and make sure it stays connected, sensor nodes can be moved around dynamically while the network is running to fill in gaps left by nodes that fail. For example, several sensor nodes near the sink may stop working because they are on active paths most of the time and consume a lot of energy in relaying messages to the sink. One way to solve this problem would be to move some redundant sensor nodes from another nearby area to replace these dead sensor nodes. Thus, replacing or reinforcing sensor nodes in highly active regions of the network with movable redundant sensor nodes can improve the overall connectivity and performance of the network. Most of these techniques are reactive in nature, i.e., they attempt to relocate nodes when a failure occurs in order to restore connectivity However, the accuracy of these techniques is dependent on the location of the void region created by the failed node, such as how far and how frequently the mobile nodes must move to reach the target area because each movement consumes a significant amount of energy. Numerous on-demand relocation techniques have been mentioned in the literature as follows. The authors of [63] proposed a distributed method for re-establishing network connectivity. The strategy makes use of mobile nodes to deal with network dis-connectivity caused due to failed nodes. It utilizes multiple sensor nodes to resume connectivity involving many cascade motions. which may lead to aggressive energy loss in many nodes. Similarly, in [64] a cascading relocation of nodes is proposed to recover connectivity in case of failed nodes. This method involves replacing the failed node with its neighbor, which then replaces by another node if its relocation caused coverage problem. Thus, this process continues until the network is completely healed. This method uses an excessive number of cascade movements, which results in a lot of movement overhead in the network, which consumes allot energy. It relies on redundant nodes in the network. The authors in [65] presented a Recovery by Invert Motion technique (RIM). In this technique, the recovery nodes moves inward toward the fail node to recover connectivity. The proposed technique relies on neighborhood information.

such that, as a result of the absence of the HEART BEAT message, neighbors will become aware of the node failure, and thus they will move toward the failed node in order to restore network connectivity. When there is a high node density, RIM struggles since it has a tendency to shift numerous nearby nodes, hence extending the overall distance covered by multiple nodes, which ultimately results into extra network overhead. PADRA, a Partition Detection and Recovery Algorithm proposed in [66] relies on the identification of cut-vertex nodes to detect network partitioning. Each cut-vertex node is assigned the appropriate neighbor to manage its failure. Thus, pre-failure, connectivity and coverage are recovered in this manner. All the nearby nodes involved in the restoration of connectivity must be moved. It results in similar neighbors moving in cascades. Due to the cascade movement of everyone involved in the connectivity restoration process, PADRA utilizes more energy. furthermore, the network coverage is not taken into account. In [67] authors proposed a technique called Volunteer-instigated Connection Restoration (VCR), which is used to restore the network's connectivity. The failed actor node's close neighbors participate in this process. Nearness is used as the selection criterion for the nearby nodes, and cascaded movement is considered. VCR cannot ensure a reduction in cascaded movement in a dynamic environment. The VCR does not consider energy level of the recovery nodes only rely on its distance to the failed node, thus it favors the closest neighbors to participate in the connectivity restoration process. As a result, the network may see more node failures as low energy nodes participate in the process of connectivity restoration. A recovery mechanism RIR ("Recovery Algorithm with Increased Robustness") is presented in [68]. The authors studied the critical nodes in the network whose survivability is important for maintaining connectivity. The failure of such a node was addressed by replacing it with a high-energy node nearby, so the strategy favors replacing a failing node with a node with the highest remaining energy. It aims to find the best set of node relocations for repairing the network topology while minimizing the movement overhead of the recovery process. By placing a node with the highest energy reserve in place of the failed node, RIR opts to reduce the risk of network segmentation caused by the threat of energy loss from the failed node. RIR orchestrated a migration of multiple nodes to recover links in order to limit the overall distance traveled by the nodes involved, thereby reducing the load on those nodes. RIR only addressed the connectivity problem while neglecting the coverage problem. The authors in [69] present a unique connectivity restoration method named "geometric skeleton-based reconnection" (GSR). GSR employs a method based on a geometrical skeleton to logically divide a network into distinct pieces. A geometrical skeleton's backbone comprises a collection of nodes having the highest connectivity to other nodes. Each segment contains a record of the nodes of the skeletal backbone: In the event of network segmentation, each segment tries to join the geometrical skeleton. To restore connectivity in partitioned wireless sensor networks, the authors in [70] introduces the OCRS-MD (Obstacle-avoiding Connectivity Restoration Strategy for Partitioned WSN Using MD-carriers) technique. This method employs a mobile data collector to collect data from the disjoint network segments. They intend to minimize movement by detecting barriers free optimum paths among partitioned segments in order to complete recovery in a timely manner. The authors in [71] presents a Survivability-Aware Connectivity restoration (SACR). When connecting subdivided segments, the data load of disjoint parts is taken into account. Between different disjoint segments, the correct position of these inaccessible segments is determined by a series of moving nodes. Immediately after tracking isolated segments, a relay partition is created for each isolated segment. This restores connectivity, mending the path from the area of interest to the sink. This technique is not suitable for mission-critical applications because the algorithm consume a lot of time in finding disjoint segments. Another method for calculating the cut vertex is described in [72]. In this method, both localized and distributed algorithms are used to detect cut and non-cut vertex nodes. The first algorithm uses a local two-hop sub-graph with knowledge of Connected Dominating Sets (CDS). The second proposed algorithm is completely dependent on the first one. Without searching the entire network, a limited distributed deep-first search algorithm is used for undetected parts of the network. This algorithm determines the states of all nodes through extensive test-bed experiments. If a CDS is present. cut-vertex nodes are detected with less energy consumption. An approach for network partition healing through movable relay node deployment is presented in [73]. The proposed solution finds the minimal number of nodes through heuristics to relocate in order to achieve connectivity. A Monte-Carlo Markov Chain (MCMC) solution to determine coverage of WSN is discussed in [74]. MCMC is used to determine the probability that WSN with multi-state nodes can efficiently send its sensed data to the mobile sink within a specified time. Deployed node use random duty cycling method to save energy. A partition healing algorithm based on node re-positioning the most optimum nodes in each partition is presented in [75]. The proposed solution use fuzzy logic for selecting recovery participant on the basis of residual energy. In [76], the authors proposed a mobile node based algorithm. The algorithm use recovery information, stored in each sensor node. This information is updated periodically to recover from network partitioning. A partition detection and recovery approach is presented in [77]. A collaborative node movement is used to cover the void region. A re-connectivity approach is proposed in [78] using mobile restoration nodes. These restoration nodes are selected from each partition and to heal the network through their movement. A partition recovery scheme using Voronoi polygon centroids is proposed in [79]. Authors exploits centroids for finding blind zone in network coverage. Authors in [80] proposed a two-phase redeployment technique for the identification of disconnected partition in a sensor field. In the first phase, a partition detection process is performed, and in the second phase, called the confirmation phase, the disconnected region is confirmed. After that, a mechanism for network reconnection is provided through mobile sensor nodes, which finds an optimized location for new nodes to be moved to reconnect the network. In [81] the re-connectivity of the isolated segments of a Mobile Robotic Sensor Network (MRSN) has been discussed. The authors investigated the obstacles and constraints during the relocation of sensor nodes to complete the broken link among different isolated network segments. Furthermore, they presented an Obstacle-avoiding Connectivity Restoration Strategy(OCRS) for mobile nodes to achieve connectivity. Devi *et al* [82] presents a relocation mechanism, for mobile sensor nodes to repair a partitioned network. They discussed the relocation of the mobile node to fill in the void space among separated segments of the network, by taking into account its degree of connectivity with the neighboring nodes. Guizhen *et al* in [83] presented a Distributed Connectivity Restoration Strategy (DCRS) to cater to network partitioning, due to the failure of critical nodes located at the cut vertex. They proposed a mechanism to identify an appropriate backup node for every critical node in the network to replace the failed node in case of network partitioning. To deal with the dis-connectivity among different segments of a partitioned network, a distributed reverse constrained recovery (RCR) mechanism was presented in [84]. They used mobile sensor node/s to act as relay node/s to recover connectivity. Similarly, in [85] mobility aware spacial sensor nodes called mobile data collectors(MDC) were proposed to overcome network partitioning. The paper presented an Obstacle Aware Connectivity Restoration strategy (OACRS). This strategy utilize, optimize obstacle avoiding spanning three mechanism to determine the minimal number of MDC nodes needed and their respective position to overcome network partitioning. The authors in [86] taken into account the realistic view of deployment region, i.e. terrain variation to determine the optimal path for mobile nodes' displacement to cover the void area which causes network partitioning.

2.1.2.3 Adjustable Transmit Power Based Solutions

Apart from aforementioned approaches, power adjustment is also used in some network recovery solutions. In case of network partition, certain authors propose usage of existing residual energy of any sensor node to accomplish data transmission. A network partition recovery algorithm through adjustable sensing range and mobile relay nodes is presented in [87]. Authors uses both node mobility and power adjustment to transmit data from the sensing device to the sink. Similarly, to address the connectivity restoration issue, the authors in [88] proposes an intelligent on-demand connectivity restoration technique for wireless sensor networks in which nodes employ their transmission range to ensure connectivity and the replacement of failed nodes with redundant nodes. The proposed method monitors the system topology and can proactively respond to node failures by relocating a redundant relay node to replace the failed node. In [89], deployment of long-distance routing sensor nodes were proposed to overcome connectivity problem among different segments of the wireless sensor network. The technique is also evaluated for Received Signal Strength Indicator (RSSI) to assure high-quality wireless links for communication. In [90], the authors targeted a spacial network partitioning scenario. In some cases due to environmental constraints such as fog, rain, etc., there is a chance of communication range shrinkage. i.e. environmental constraints can reduce sensor communication range which may result in network partitioning. The proposed technique addresses the problem by adjusting the sensor communication range.

2.1.3 Collaborative Beamforming in WSNs

In order to establish re-connectivity of the partitioned network through non-conventional means such as beamforming, we investigated some State-of-the-art cooperative beamfroming mechanisms in wireless sensor networks as follows. In recent years, cooperative communication through beamforming has attracted the attention of numerous researchers. Many schemes have been proposed for beam formation [91–98] and its effective use in energy efficiency [99–104], power optimization [105] and node localization [106]. However, to the best of our knowledge, the use of cooperative beamforming in the re-connectivity of abandoned general-purpose nodes with no extraordinary smart antenna facility, trapped in a disconnected cluster of a wireless sensor network, has not been incorporated so far. The authors, in [94, 96], proposed the idea of cooperative communication through the nodes equipped with the smart antenna. They introduce the methodology of collaborative interaction between nodes to form Multiple-Input and Multiple-Output (MIMO) link for enhancing diversity and data rate of the network. The technique uses the intelligent adaptive antenna system instead of Omnidirectional antennas. The distribution of wireless sensor nodes in an area of interest with their cooperative beamforming is discussed in [92]. The paper investigates the challenges such as the orientation and placement of sensor nodes for their efficient cooperative beamforming. It uses a Gaussian distribution function to model the spatial distribution of sensor nodes. In [93], the effectiveness of the beam formation has also been studied for different sensor node distribution models. The paper shows that collaborative beamforming provides better performance when sensor nodes' deployment follows a Gaussian Probability Density Function(PDF) as compared to uniform PDF. The authors in [97] proposed a method to determine the minimum connectivity probability for a given transmission range concerning any nodes' density with varying antenna beam-width of a smart antenna system mounted on special-purpose wireless sensor nodes. With this method, they found the minimum number of smart nodes with various beam widths required in a given region. The phase synchronization of the transmitted waves over time-varying channels has been discussed in [98]. The authors propose a method which uses continuous feedback from the receiver to optimize the phase of transmitting waves to form an effective beam. In [91], the authors analyze the behavior of cooperative beamforming on different number nodes to several nodes involved and type of antenna installed on sensor nodes. They observe that as the number of nodes involved in beamforming increases, the beamforming becomes more prone to errors, power synchronization is being an issue.

In [100], a convex optimization framework is used for collaborative beamforming to increase the network lifetime and to satisfy a predefined quality of service(QoS). Similarly, in [104], a cooperative beamforming mechanism is investigated for energy-efficient communication. The paper also proposed a mathematical framework for the selection of an appropriate number of nodes, required to perform beamforming. The result shows that with an increase in the number of cooperative nodes, the energy efficiency is improved for cooperative transmission, but more energy would be needed for sensor selection and beamforming. In [101], a scheduling algorithm is proposed for data transmission using beam formation. In this algorithm, the network is divided into clusters, and the nodes perform each round of cooperative transmissions through beamforming via their cluster heads. Thus reducing the energy consumed on data sharing and thus improves the overall lifetime of the network. The experimental results are presented, which shows that the beam formation achieve more transmissions than direct or multi-hop communications. A convex optimization framework which optimizes the collaborative and distributed beamforming is presented in [99]. This methodology also improves the network lifetime and optimises battery consumption of the WSN nodes.

In [105], the node selection for cooperative beamforming is made in coherence with a uniform circular array. The authors suggest, that while selecting sensor nodes for cooperative beamforming, uniformity of the nodes in a circular array should be taken into consideration. The idea of cooperative communication for wireless sensor network node localization is presented by [106]. The authors propose a method to solve the localization issue through collaborative communication among the nodes. The nodes calculate its location. In this method, the nodes share their positions in the transmitting messages, and the receiving node can find its location with the given information.

Summarizing the above discussion, the existing methods in the literature for reconnecting a disconnected network rely primarily on topological changes in the network, which is both time and resource consuming. Therefore, these techniques cannot be used for valuable time-sensitive data. There is a higher possibility of losing important real-time data collected by the disconnected sensors. Furthermore, the existing techniques cannot be used to form a viable network for reliable detection in time-sensitive situations such as safety and risk environment monitoring in mining, industry and forest fire prevention, etc. The significance of network connectivity in wireless sensor networks cannot be overstated. Despite numerous research endeavors to tackle the challenge of network partitioning and re-connectivity, it remains an ongoing area of investigation. With wireless sensor networks becoming more pervasive and being employed in increasingly crucial applications, there arises a pressing demand for more resilient and dependable solutions to effectively address the issues of network partitioning and re-connectivity. Advancements in this field are crucial to ensure uninterrupted data transmission, enhance system reliability, and support the seamless operation of critical applications.

2.2 Gap Analysis and Research Motivation

The main purpose of wireless sensor networks is to sense their environment and transmit this data to a sink node. Therefore, network connectivity is one of the primary design goals of any wireless sensor network. Existing research shows that the failure of a sensor node at a key location leads to disruption of network connectivity and eventually to the partitioning of the network, as shown in Figure 2.1. Consequently, there are two basic circumstances that can lead to network

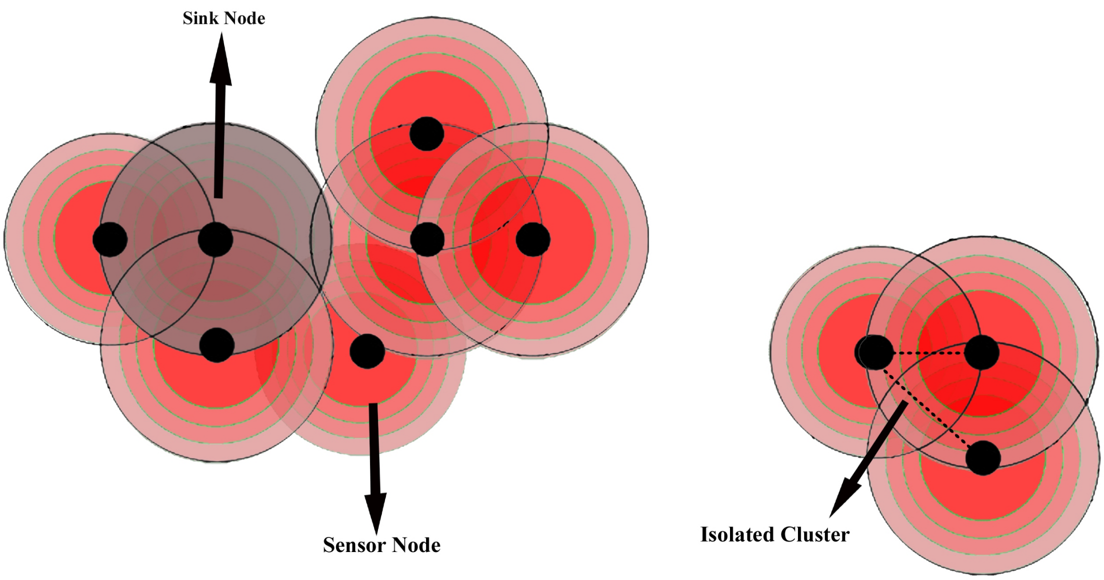

FIGURE 2.1: Network Partitioning

partitioning in a wireless sensor network, namely, 1) partitioning that occurs in the sensor node deployment phase as a result of uneven deployment or loss of a node during deployment [107]. 2) partitioning due to a critical node failure during the operation of a wireless sensor network [38, 108]. The existing literature mainly focuses on partitions caused during operational network due to node failure [109]. One of the major reason for node failure during operational network is energy depletion, apart from hardware failure [110, 111]. such that majority of the sensors are equipped with limited non-rechargeable power sources [2, 3]. Most of the available methods rely on neighbor based techniques where the basic assumption is the network in considered initially fully connected in order to maintain a list of connected neighbors and monitor the status of their respective link through period messages. However, the partitioning caused by the void regions created during deployment is not considered, as discussed in section 2.1.2. Since the sink node has no clear knowledge of the status of these alive nodes that are partitioned and the sensor nodes are unable to contact the sink in a conventional manner.

Moreover, most available methods for reconnecting a partitioned network rely on topological changes within the network. They attempt to solve the re-connectivity problem by relocating mobility-enabled sensor nodes to the uncovered region or by deploying additional relay nodes to provide connectivity. Both conventional methods of partition healing require additional resources and are thus time-consuming. Therefore, these methods cannot be used for time-critical applications that rely on real-time information. There is a greater risk of losing important real-time data collected by the disconnected sensors. In addition, environmental conditions may make it impossible to relocate or deploy additional sensor nodes to restore connectivity, especially in hazardous environmental monitoring, security applications, industrial and forest fire prevention etc. In such time-critical situations, a viable method is needed to restore connectivity using available resources.

The above situation especially becomes noteworthy when the partitioned cluster contains a large number of nodes with sufficient power and a handful of data that are unable to connect to the main network individually. Nevertheless, in the above-mentioned conventional methods, partition re-connectivity is not feasible in real-time. Thus, in order to establish a communication link to forward important data contained/collected by the nodes in the disconnected cluster to the sink, the idea of the distributed collaborative beamforming approach can be utilized.

In a nutshell, this research focuses to address the following research questions..

- How to detect partition created at deployment phase without utilizing network state information.

- How to recover partition with available resources without disturbing the network partition.

- How to ensure connectivity in a timely manner to minimize critical data loss in real time application.
- How to determine the number of alive nodes in a disconnected partition, in order to initiate re-connectivity.

2.3 Problem Statement

Based on the discussion in section 2.2, we have concluded that the following major issues need to be addressed to overcome the above-mentioned limitations in the existing partition detection and recovery mechanisms.

1. There is a need for improved partition detection mechanism that covers all partitioning scenarios, such as partition occurrences at both deployment and operational phases due to critical node failures, and differentiates between alive and dead partitioned node/s.
2. An optimized partition detection and re-connectivity mechanism is required, which can perform partition healing using the existing resources in a timely manner and retrieve time critical data trapped inside an isolated partition.

2.4 Research Methodology

The research methodology outlined in this study aims at starting from the investigation of the impact of nodes' failure on the network connectivity in wireless sensor networks (WSNs), specifically focusing on network partitioning. The outcome of the investigation will then lead us to choose the best methodology to identify hidden network partitions. In this regard, the conventional and non-conventional techniques will be explored. Since conventional methods have already been investigated well and have got their intrinsic drawbacks; therefore, research will be done to design a robust and fault-tolerant system that addresses data critical WSN applications by utilizing non-conventional methods like cooperative beamforming techniques to retrieve time-critical data from isolated clusters of nodes. Lastly, since preemptive techniques are also needed in WSNs to avoid network partitioning, therefore, we intend to devise a partition avoidance mechanism by utilizing energy visualization techniques. The objective is to proactively prevent network partitions from occurring by analyzing the energy distribution within the network. By identifying areas of potential partition formation, appropriate strategies and mechanisms will be developed to avoid or mitigate partitioning scenarios. This will contribute to enhancing the stability and longevity of the WSN by minimizing the occurrence of network partitions. Throughout the research process, data collection will be conducted using simulation tools and possibly real-world deployments. The performance of the proposed approaches will be evaluated based on metrics such as partition detection time, and energy consumption. The research findings will be analyzed, interpreted, and discussed to draw meaningful conclusions regarding the impact of failed nodes on network connectivity, the efficiency of the proposed methodologies for partition detection and reconnection, and the effectiveness of energy visualization techniques for partition avoidance.

2.5 Research Contribution

This dissertation discusses the challenges of identifying the network partitioning and restoring connectivity in wireless sensor networks based on time critical data.The contributions of the proposed research to the existing body of knowledge are thus summarized as follows:

1. In Chapter 3, we present the first contribution of this research, which is a novel mechanism for identifying network partitioning and re-connection using additional relay nodes. This proposed system identifies network partitioning by collecting neighborhood information in a traditional manner. Moreover, we have developed an efficient technique based on the Receive

Signal Strength Indicator (RSSI) to deploy relay nodes and restore connectivity in a realistic 3D environment. The primary objective of our research is to identify the maximum number of network partitions while improving the time efficiency of partition identification through the application of graph theory.

2. The second significant contribution of this research is presented in Chapter 4, where a novel partition identification and connectivity restoration mechanism for wireless sensor networks is developed through cooperative beamforming. The proposed technique, along with its advantages and limitations, is thoroughly discussed.

3. As the third contribution, comprehensive computer simulations are conducted to analyze and compare the performance of the proposed mechanism with existing methods. The simulation results provide insights into the effectiveness and efficiency of the proposed approach, and demonstrate its superiority in terms of various performance metrics.

4. In Chapter 5, we present the fourth contribution of this research, which is an energy visualization technique for wireless sensor networks. This technique enables network managers to predict potential node failures due to energy constraints, detect energy holes, and identify potential network cuts or partitions. With this information, network managers can deploy additional sensors in a timely manner to maintain connectivity and prevent partitioning. The proposed technique is evaluated through computer simulations, and its effectiveness is demonstrated by comparing it with existing approaches.

Chapter 3

Partition Identification and Redeployment Algorithm

3.1 Introduction

The performance of the WSN depends entirely on the fact that the sensor nodes remain connected to the sink node to exchange data. Since there may be some nodes that are far away from the sink node and need many hops through their neighboring nodes to transmit their data to the sink node; therefore, they rely on certain intermediate nodes to ensure their connection to the sink [112]. The loss of a key sensor node that is in the communication path can result in a number of still-living sensor nodes being unable to reach the sink and no longer being part of a functional network. The failure of a single node or multiple nodes can affect the entire network and can introduce certain cuts and coverage holes, which may leads to network partitioning [16].Thus resulting in the failure to achieving required objects for which it is being design, such as monitoring, surveillance etc [18]. Identifying these partitioned clusters of nodes is a difficult task. If all communication paths to the sink are broken, there is no explicit way for the sink to know about these clusters of nodes that may still be alive and may have important data to send. Normally, it is assumed that they are dead. However, in many cases they may not be dead; therefore they may be reconnected to the sink. This will not only extend the overall life of the network but also increases the network performance.

Once the sensor network is operational, many essential nodes may perish due to energy depletion if these nodes are positioned at cut vertex points as relay nodes to provide connectivity to the sink node positioned in the network's core.In this way, network leads to partitioning. Many current solutions, including HCCI [40], employ a probabilistic model to identify network partitions and assume that any node failure results in a partition, which may not always be the case. A deterministic approach is required to determine whether the failed node actually caused the partition. In contrast, DPDRU [38] relies on pre-identified cut vertex nodes and assumes that a partition has occurred whenever a node belongs to the cut-vertex nodes is reported to the sink to be dead. However, the problem with this approach is that every failed node can alter the network topology, which changes the cut vertex nodes, resulting in a significant amount of control message overhead needed for the system to recalculate the cut vertex nodes each time a node fails.

Our proposed approach Partition Identification and Redeployment algorithm (PIRA) localizes the issue and does not depend on pre-calculated cut vertex matrices, greatly reducing the complexity. It attempts to determine the partitions deterministically and offers a re-connectivity mechanism by optimizing relay node deployment through the use of Receive Signal Strength Indicator (RSSI).

The proposed algorithm is a two-step approach. First, we detect network partitioning caused due to sensor node failures at critical locations. Secondly, we propose an optimized relay node redeployment technique to reconnect these isolated partitions to the main network, which contains a sink node.

The chapter is further subdivided as follows. Section 3.2 describes the system model, whereas Section 3.3 discusses the startup step. The identification of partitions is explored in section 3.4, and partition recovery using efficient sensor redeployment is covered in section 3.5. whereas section 3.6 deals with results and discussion. Section 3.7 conclude the chapter.

3.2 System Model

We consider a randomly deployed wireless sensor network that consists of static sensor nodes with their IDs listed in a vector V represented as

$V = \{v_i \in V \ where \ i = 1, 2, 3...n\}$. The sensor nodes are equipped with a GPS system that enables them to determine their own location. All nodes are identical in terms of their energy and communication capabilities. The nodes are expected to monitor their environment within their sensing range and to transmit the sensed data to a central repository called the sink. The sink is a node with advanced computing capabilities and high processing capacity. It is responsible for data collection and management of the sensor network. It is assumed that all sensor nodes in the network have access to the sink node via direct links or multi-hop links. The sensor nodes are considered connected or one-hop neighbors if they are within the communication range of each other, such that $D(v_i, v_j) \leq R$. Where D represents associated distance among nodes v_i and v_j and R is the communication range. A Beacon or HELLO message is used to obtain one-hope neighbor information. Furthermore, the sink node will maintain a list of all the alive connected sensor nodes V_s as active nodes , such that $V_s = \{(v_i | \forall v_i \in V\}$.

3.3 Initialization Phase

Once the wireless sensor network is deployed. The initialization phase will kick in, and network configuration will start. The detail of the initialization phases for sensor node and sink are given in Algorithm 1 and Algorithm 2, respectively.

1. Initially, each sensor node computes its one-hope neighbor list, $N(v_i)$, using a $HELLO$ message exchange.

$$N(v_i) = \{v_j | i \neq j \text{ and } v_j \text{ is in communication range of } v_i\} \tag{3.1}$$

2. The sensor nodes send a compound message, $msg(N(v_i), l_i, v_i)$, to the sink. This message contains the updated one-hop neighbor list $N(v_i)$, location information $l_i = (x_i, y_i, z_i)$ and the node ID v_i.

Algorithm 1: Sensor Node Initialization Phase

$Broadcast(\text{"}HELLO\text{"})$ `// Send hello msg to all nodes`
$N(v_i) \leftarrow response(v_j)$ `// Receive response from all one hope nodes`
$Setup\_msg \leftarrow Create\_msg(N(v_i), l_i, v_i)$
$Setup\_msg\_sink(Setup\_msg)$
$Wait\ for\ Ack\_sink\ \ for\ \ 2RTT$
if *Ack_sink is recieved* **then**
    $update\_status(active)$ `// active means node is connected to sink`
else
    $update\_status(partition)$ `// partition means node is not connected to sink`
end

---

3. The sink node on reception of all the setup information from the corresponding sensor nodes will send an acknowledgement message *Ack_sink* to the sensor nodes in order to validate their connectivity to the sink node. All those sensors node that receive the acknowledgment from sink will update their status to active (connected nodes).

Algorithm 2: Sink Initialization Phase

Input: $V = \{v_1, v_2, \ldots v_n\}$: Total Sensors nodes
Output: G : Undirectional Graph
$V_s \leftarrow \phi$ `// List of all alive connected nodes`
$V_d \leftarrow \phi$ `// List of dead nodes`
foreach $v_i \in V$ **do**
    $Setup\_msg \leftarrow Listen(v_i)$ `//` v_i `location, neighbor list, ID`
    if *Setup_msg is Rec from* v_i **then**
        $V_s \leftarrow V_s \cup (Setup\_msg)$
        $Send(Ack\_v_i)$
    else
        $V_d \leftarrow V_d \cup v_i$
    end
end
$G \leftarrow construct\_graph(V_s)$ `// Construct Undirectional graph from` V_s

---

4. In addition, the sink node will perform the following tasks after receiving the status information from all the active sensor nodes in the network.

    - The sink node will maintain the list of all connected alive nodes refer as active nodes V_s whose setup message being received on the sink, $V_s = \{v_i | v_i \in V$ Nodes who reported to sink $\}$

      It then compares the received node information with the existing deployed node list V and filter out the non-reporting nodes V_p initially assume to be alive but not being able to reach the sink node.

$$V_p = V - V_s \tag{3.2}$$

    - The sink node will maintain a connected graph $G(V_s, E_{i,j})$ based on the information received from the sensor nodes. Where V_S represent the connected alive nodes in the network, and $E_{i,j}$ represent the edge between the nodes i and j, which are in direct communication to each other, i.e. one-hop neighbors.

$$E_{i,j} = \left\{ \begin{array}{ll} 1 \text{ if } v_j \in N\left(v_i\right) \;\; \forall i = \left\{1, 2, 3, ...n\right\} i \neq j \\ 0 & \text{otherwise} \end{array} \right\} \tag{3.3}$$

3.4 Partition Identification Phase

In a wireless sensor network, all nodes follow a multi-hop communication pattern to transmit data to their intended destination.They depend on intermediate nodes laying at path to destination for their communication. Thus, there is a considerable probability that a critical bottleneck node on the communication path would perish due to rapid energy depletion, resulting in network partitioning. The proposed approach for restoring network connectivity and coverage begins in the same way as worked out by Ranga *et al.* [38]. Our proposed solution assumes that all sensor nodes are aware of their one hop-neighbors and will report any node failures in their corresponding neighborhood to the sink node. Ranga *et al.* employ a predetermined cut-vertex list on the sink to decide whether or not this node failure qualifies for network partitioning. The issue with this technique is employing a pre-calculated cut-vertex list to determine whether a failing node is eligible to participate in network partitioning. For example, the cut-vetex list is not a constant list calculated once during the start-up phase, but depends on the topology of the network, so that each failure of a node will most likely result in a new set of cut-vertexes. Consequently, every time the network topology changes, the system/sink must constantly recalculate the cut-vertex nodes of the entire network. This is a relatively expensive task to compute. In our proposed method, we restrict the problem to the area of the network where the failure of a node has occurred and develop a reactive method to identify the partitioned cluster of nodes, which is described in Algorithm 3.

Algorithm 3: Partitioned Identification in operational WSN

//Operation performed on Sensor node v_i
Send periodic hello msgs to its Neighbors $N(v_i)$
Wait for ack `// wait for 2 consecutive acks`
if *ack_not_rec* **then**
    Consider $v_d \in N(v_i)$ as dead
    Report v_d to Sink
    $Update(N(v_i))$
end
//Operation performed on Sink
Receive report of v_d as dead node
$V_s \leftarrow V_s - v_d$ `// remove` v_d `from active nodes list`
$V_r \leftarrow \phi$ `// initialize` V_r `as empty`
foreach $v_i \in N(v_d)$ **do**
    if v_i *report* v_d **then**
        $V_r \leftarrow V_r \cup v_i$
    end
end
$\widetilde{N}(v_d) \leftarrow N(v_d) - V_r$
$G \leftarrow \bigcup_{\forall v_j \ reachable \ \widetilde{N}(v_d)} N(v_j)$ `// G represent partition equation:3.5`

Once the network is operational, the sensor nodes use regular heartbeat messages to monitor their one-hop neighbours. If it does not receive two consecutive heartbeat messages from its neighbour, it declares the node dead v_d and notifies the sink node. Thus, it can be assumed that the sink receives information from its active neighbours about the failure of the failed node. The sink node waits for a certain time until it receives all failure notifications and maintains a list of nodes reporting failures as $V_r = \{v_i | v_i \in N(v_d) \text{ reporting dead node}\}$. Since the sink node has already have information about the one-hop neighbourhood of each node received in the initialisation phase, the sink node will generate a partition graph $\widetilde{G}$ after receiving all failure reports.

Let V_r be the set of active one-hop neighbours of the dead node v_d and $N(v_b)$ be the neighbour list of the dead node v_b computed in the initialisation phase. The unreachable one-hop neighbours $\widetilde{N}(v_d)$ are given as

$$\widetilde{N}(v_d) = N(v_d) - V_r \tag{3.4}$$

The network partition sub-graph $\widetilde{G}$ can be given as, let v_d is the dead node then the partitioned graph $\widetilde{G}$ will contain all the nodes that are reachable from v_d, here $\widetilde{N}(v_d)$ represent the unreachable(partitioned) one hope neighbours of v_d while $N(v_j)$ represent the set of node that are reachable from v_d indirectly.

$$\widetilde{G} = \bigcup_{\forall v_j \ reachable \ \widetilde{N}(v_d)} N(v_j) \tag{3.5}$$

When the resulting subgraph is completely isolated from the rest of the network and has no edge/connection to the main network. It is considered to be partitioned.

3.5 Connectivity Restoration in 3D Environment

The redeployment of the relay nodes is a complex problem in itself. To perform efficient redeployment of relay nodes to achieve connectivity of partitioned networks, we need to consider the real environmental factors. To deal with the problem of deployment in uncertain terrain, elevation factor is incorporated in order to obtain a realistic deployment map. The problem becomes even more complex when multiple nodes fail, resulting in a large uncovered area, and we need to deploy

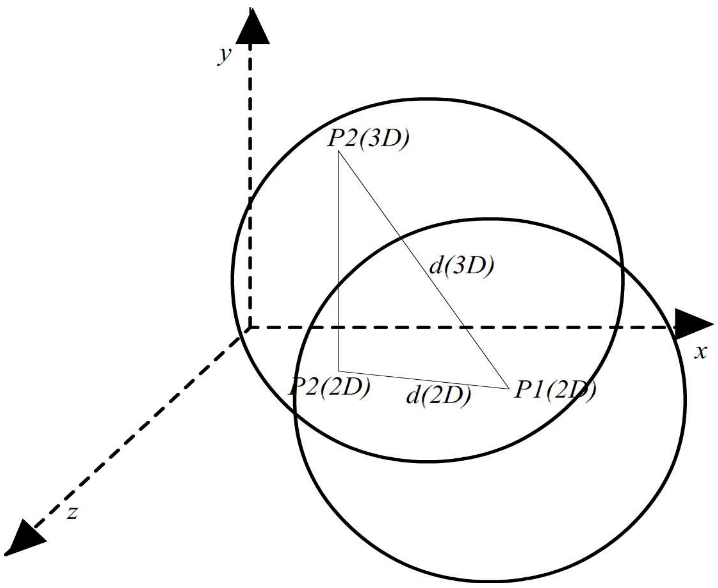

FIGURE 3.1: Communication range of nodes in 2-D and 3-D planes such that $d_{3D} > d_{2D}$

multiple relay nodes to reconnect the partitioned network. In the literature, most researchers consider flat surfaces for sensor deployment. However, in the real world environment; the field may consist of irregular terrain with obstacles between sensor nodes. Obstacles may include elevation factor, water, rocks, forests, etc. Due to these factors, the sensing model of the relay nodes may be affected. If we consider the case of elevated terrain, when 2-D distribution schemes are used, the direct communication paths between additional relay nodes may be affected by the elevation. For this reason, current 2-D distribution schemes in elevated terrain do not meet the deployment goals in terms of coverage and connectivity. The problem is illustrated in Figure 3.1. In the 2-D plane, two nodes are installed at points $P_{1(2D)}$ and $P_{2(2D)}$. In the 2-D plane, both nodes appear to be connected, but in reality point P_2 is located at a certain elevation, outside the communication range of point P_1. By incorporating the terrain factor, the added elevation will shift the node at a point $P_{2(2D)}$ in 2-D plane to another point $P_{2(3D)}$ in 3-D plane, thus separating them with distance d_{3D} instead of d_{2D} and making them disconnected. Thus, considering a 3D realistic environment for relay node deployment, we come across two basic problem scenarios.

- The exact locations of the nodes in 3D plane affect the separation distance among them, but they may still co-exist in the line of sight of each other.

- The exact locations of the nodes in 3D plane may be the same as that in 2D plane. This will not affect the separation distance among the nodes, but an elevated blockage or hurdle may cause signal deterioration due to multipath fading.

Therefore, to cope with the above-mentioned problems, we propose a two-step technique to identify the appropriate location for relay node; i.e 1) relay node deployment 2) relay node optimization.

3.5.1 Relay Node Deployment

In the first step, the communication gap between two partitions is investigated to determine the appropriate locations to deploy additional nodes. To connect the partitions with an optimized number of nodes, the closest pair of nodes between two partitions is first determined, and additional nodes are installed between the closest nodes.

For example, let $P_{1(3D)}$ and $P_{2(3D)}$ be the nearest nodes between the two disconnected partitions. The number of additional relay nodes K required to connect these partitions can be determined by the following relation.

$$K = ceil(D/R) - 1 \tag{3.6}$$

Where K is the number of nodes to be deployed, R is the communication range, and D is the minimum distance between the nearest nodes of two partitions. After determining the number of additional nodes to be deployed, we determine the distance d between the edge node of one partition and the additional relay node on way to the nearest edge node of the other partition as; The second relay node shall be placed at distance d from the first relay node. This process will continue till all K additional nodes are placed between the two nearest edge nodes of the two partitions.

$$d = D/(K+1) \tag{3.7}$$

3.5.2 Relay Node Optimization

In the optimization phase, we further optimize the redeployment by considering real-world environmental conditions, such as terrain elevation, hurdles, and blockages between deployed relay nodes. As mentioned earlier, there is a high probability that these hurdles between relay nodes may restrain them of meaningful communication due to fading effects. Even if the nodes are in communication range with respect to their minimum distance, they still may fail to provide proper connectivity. To tackle such situation, we propose a technique to optimize relay nodes deployment through Received Signal Strength Indicator (RSSI).

3.5.2.1 Euclidean Distance Calculation

Since each relay node has its specific communication range. Therefor for effective communication among two neighboring nodes, it is important that the separation distance between the respective nodes is less than or equal to the communication range of the sensor node, i.e.

$$d_{ij}^{cal} \leq R \tag{3.8}$$

Where d_{ij}^{cal} is the Euclidean distance between two communicating sensor nodes. There is a fair chance that with the introduction of elevation factor, this separation distance will be disturbed, and the previously connected nodes will move out of the communication range.

3.5.2.2 Distance Estimation through RSSI

In wireless communications, the received signal strength is a key indicator for determining the overall link quality, which is essential for efficient communication.

The distance between two nodes can be determined by measuring the received signal intensity of the incoming Radio signal. In order to optimize the deployment of relay nodes for improved connectivity, we took into account both the RSSI and the Euclidean distance. After 3D deployment with a 2D deployment method, all relay nodes are required to communicate its neighbor's RSSI along with their IDs to the sink for registration. On the basis of RSSI information received from all relay nodes at the sink, distance is calculated and stated as distance estimated d_{est}.

$$RSSI \propto 1/d_{est}^{n} \tag{3.9}$$

Where d_{est} is the estimated distance and n is the path loss exponent, which depends upon environment and remains constant. n can be taken as 3 for a suburban environment. Thus, comparing distance estimated through RSSI and Distance calculated through Euclidean Distance, we come across two basic scenarios based on the available information.

1. Distance estimated will be almost equal to distance calculated.

$$d_{ij}^{est} \cong d_{ij}^{calc} \tag{3.10}$$

   It means both the nodes are located in line of sight and there is no effect on signals due to blockage or multipath fading etc. Which mean a strong communication link exist between the relay nodes, and we do not need to deploy any additional node.

2. Distance estimated is greater than distance calculated.

$$d_{ij}^{est} > d_{ij}^{calc} \tag{3.11}$$

   It may be assumed that there is no line of sight communication between the nodes, it may be due to either the nodes being located at different elevations or a hurdle resides in between them. It can be concluded that the antenna orientations do not allow for favorable LOS communication, so there is a high probability that no meaningful communication exists or very weak, error-prone NLOC communication exists between the nodes due to multipath signals reflected from objects in their vicinity.

Thus, if we want to ensure that our communication is reliable and error-free, we need to deploy some additional relay nodes to combat the fading effects and ensure robust connectivity. The technique proposed to optimize the relay nodes ensures that connectivity is maintained by a relatively small number of additional relay nodes.

3.6 Results and Discussion

The performance of the proposed technique is evaluated based on several parameters such as the number of detected partitions, the duration of partition detection, and the energy consumption of the sensor nodes. Extensive simulations were performed for the evaluation using a simulation scenario with different typologies containing 100 to 500 nodes randomly distributed in an area ranging from $400m \times 400m$ to $1000m \times 1000m$. The initial deployment is random, and each node has a constant communication range and are homogeneous in nature. The simulation setup is shown in Table 3.1. The simulation was run several times to obtain average results for the above-mentioned parameters. Since we target time-critical applications, the partition identification time is an important factor, along with the number of identified partitions. Energy consumption also become an important parameter if the sensor nodes have very limited battery reserves, and they are supposed to last till the end of application. The obtained results are then compared with two notable recent partition identification methods, DPDRU [38] and HCCA [40]. Figure 3.2 represents the partition detection time. The findings indicate that the proposed algorithm PIRA performed significantly better than DPDRU and HCCA. As previously mentioned, DPDRU uses a predetermined list of cut vertices to identify network partitioning, In graph theory, identifying cut vertex is a computationally intensive process, and the problem becomes more complex as the network size increases. Similarly, any change in network topology

Table 3.1: Simulation Setup

Simulation Parameter	Value
Area	400x400 to 1000x1000 m^2
Simulation Tool	MATLAB R2021a ®
Number of Nodes	100 to 500
Packet Size	32 bytes
Data Rate	250 kbps
Communication Range	50m to 150 m
Node Distribution	Uniform Random
Number of Simulations Rounds	10

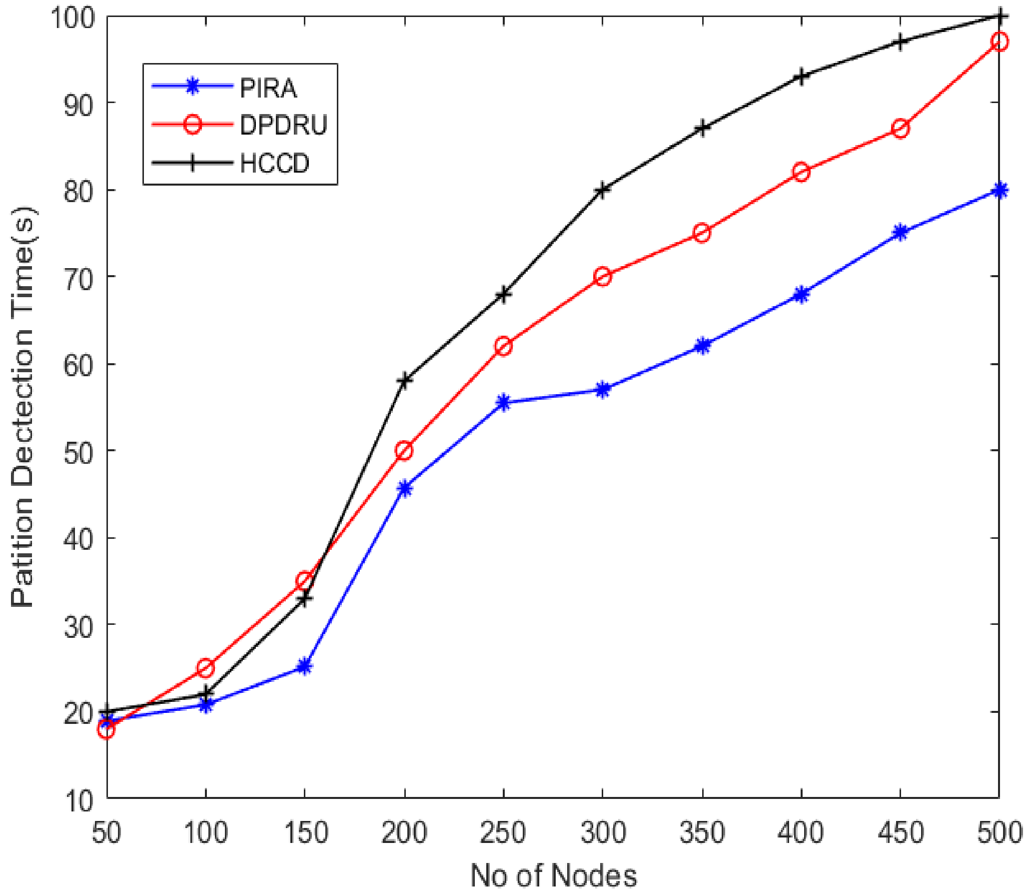

Figure 3.2: Partition Detection Time versus Number of nodes.

results in a new set of cut vertex nodes. Therefore, it is obligatory for the sink node to recalculate the cut vertex list for every failed node. In a similar fashion, HCCI rely on an iterative algorithm to identify the partitions, taking more time.In contrast to DPDRU and HCCI, our suggested method restricts the problem to the

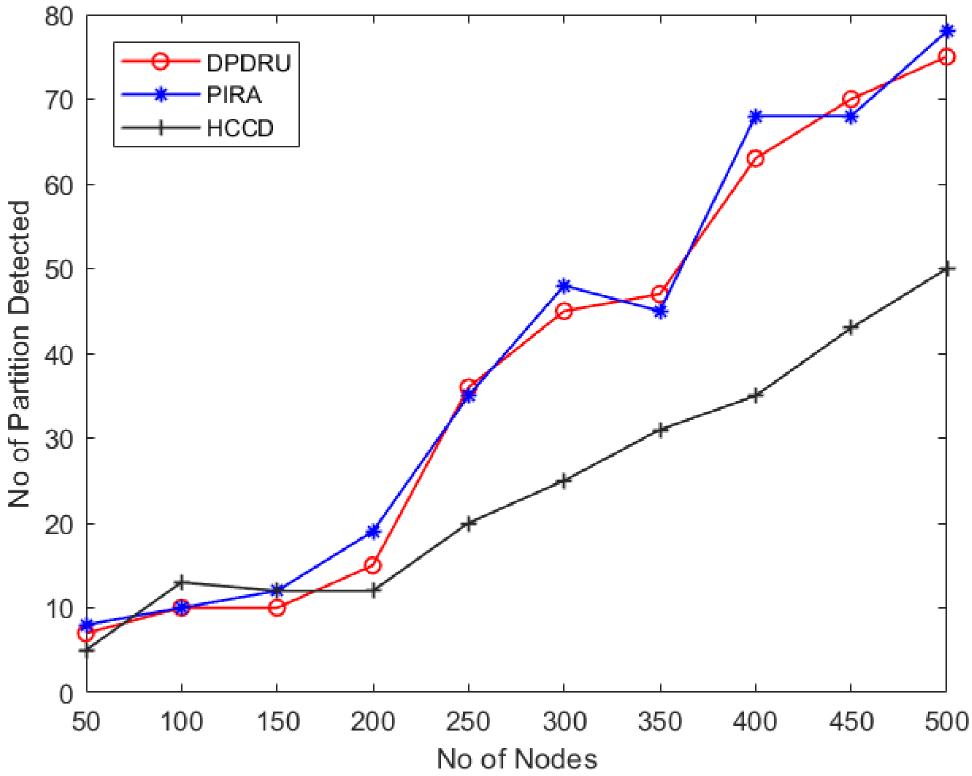

FIGURE 3.3: Number of Partition Detected versus Number of nodes.

area of interest and identifies partitions using just local neighbour information. It considerably reduces the time required to identify partitions. Figure 3.3 depicts the number of partitions isolated as a result of the failure of nodes located in critical places. Compared to the HCCI, it can be shown that our proposed system did fairly well. This is due to the fact that our proposed method PIRA is a deterministic proactive approach to identifying partitions, and that several neighbour nodes are involved in the detection process, which decreases the likelihood of false positives. HCCI, on the other hand, employs a probabilistic model to detect network partitioning. PIRA exhibits the same behaviour as DPDRU due to the fact that both techniques use neighbour node information for partition discovery. However, our proposed approach gets more efficient, as the network grows larger. The comparison of energy consumption is depicted in Figure 3.4, as our proposed approach involves fewer nodes in partition detection than DPDRU and HCCI. Only the neighbours of failed nodes communicate with the sink. The sink node is

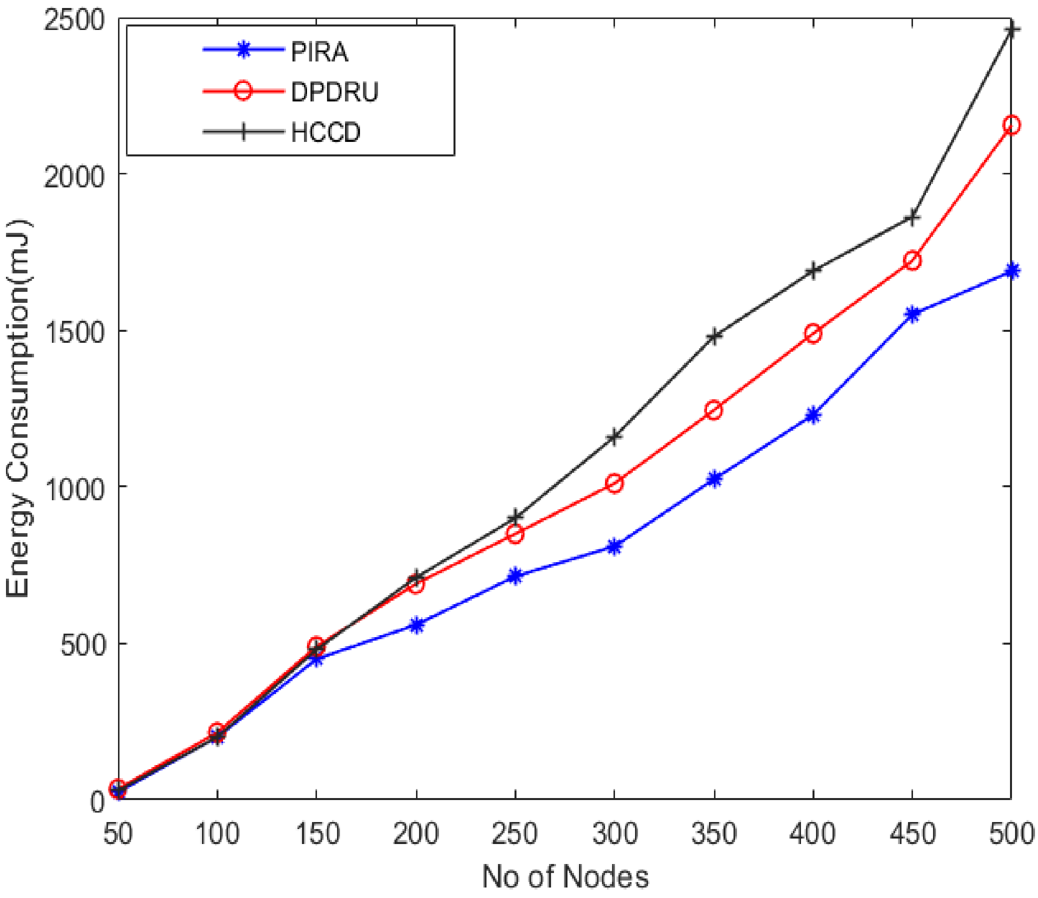

FIGURE 3.4: Energy Consumption versus Number of Nodes.

not required to recursively query the network for cut vertex list generation, which would result in additional communication overhead and increased energy usage. This creates a difference of consumption energies when the proposed algorithm is compared with DPDRU and HCCI. Therefore, It has also been noticed that as the number of nodes increases, the difference in energy consumption also increases, as DPDRU deals with the problem at the network level by recursively computing a cut vertex list for every topological change in the network caused by failing nodes. However, our proposed method is unaffected by the size of the network. It depends simply on the number of nodes adjacent to the failed nodes. In this case, only sensor nodes inside the affected area participate in the process of partition identification.

Furthermore, the connectivity of each deployed relay node with respect to the

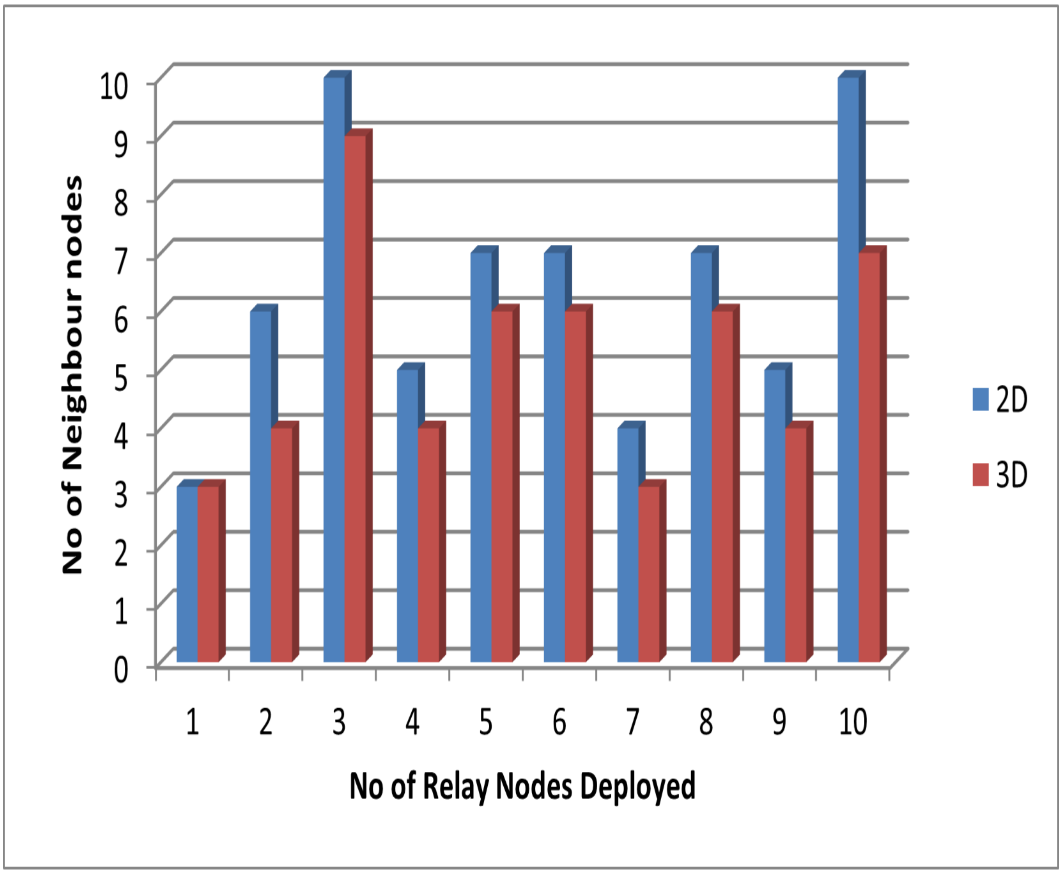

FIGURE 3.5: Number of Neighbor Nodes in 2-D and 3-D plane

neighboring nodes is compared in 2-D and 3-D planes to ensure connectivity. At first, it can be seen in Figure. 3.5

that for each node in 2-D plane, the number of surrounding connected nodes is more than 3-D plane. In the 2-D plane, the network was misunderstood as highly connected, although it was not, when evaluated in the realistic 3-D environment. This is due to the fact that for 2-D plane, the elevation factor was not taken into the consideration.

3.7 Conclusion

In this chapter, we have proposed a partition identification method based on the traditional neighbourhood information collection. Any failure of a sensor node due to energy depletion or other environmental conditions is detected by its neighbouring active nodes, which report this to the sink. The sink then determines whether the failed node caused the partition based on the information received during the initial setup phase and the information received from the neighbouring reporting nodes. No additional messaging overhead is required in the proposed algorithm. Most existing solutions, such as HCCI, use a probabilistic model to determine network partitions and assume that any node failure causes a network partition. Similarly, other approaches such as DPDRU use additional control messages to handle false positive situations. The proposed technique PIRA outperformed when compared with two of the existing techniques HCCI and DPDRU in terms of number of partition detected, the duration of partition detection, and the energy consumption of the sensor nodes. The proposed technique not only detects network partitioning in the shortest possible time without posing any additional message congestion, but also leads to significant reduction in energy consumption.

In addition, we have developed an efficient connectivity recovery technique for relay node deployment based on a real $3D$ scenario. We used the Receive Signal Strength Indicator to detect terrain variations, such as blockages between neighboring sensor nodes, which can cause fading and disrupt the range of sensor communication. We intelligently used the distance calculated via RSSI to optimize relay nodes node deployment.

Chapter 4

Partition Identification and Reconnectivity Through Cooperative Beamforming

4.1 Overview

The intensive study of antenna radiation patterns and beamforming mechanisms in the literature [22, 113] led us to the idea of exploiting the directional power dissipation property of antennas to achieve long effective communication range. A collection of nearby nodes equipped with single omnidirectional antennas can effectively increase their communication range by cooperatively combining their transmission power to form a single directional high-power beam [114]. Such a cooperative beamforming method can be used by a cluster of disconnected nodes to increase the transmission range, thereby skipping the dead hops and reaching out for alive relay nodes to communicate with the sink. In our proposed scheme Partition Identification and Reconnectivity through Cooperative Beamforming(PIRCB), we exploit the above-mentioned approach to perform partition identification and develop reconnectivity of the partitioned nodes to the sink. Since it is a cooperative beamforming solution; therefore it is not applicable to individual isolated nodes. The results show that the proposed solution is more favorable for a large number of isolated nodes. The proposed approach is helpful in dealing with partition created at the time of network deployment, as will as with traditional partitions created during the operation of the network due to nodes failure.

The rest of the chapter is organized as follows: Section 4.2 present the system model, while Section 4.3 discusses the partition detection mechanism. whereas, section 4.4 gives details of proposed cooperative beamforming. In section 4.5 we explain the procedure for searching nearby relay node and reconnection phase. Section 4.6 discusses various results to evaluate the proposed mechanism, while section 4.7 concludes the chapter.

4.2 System Model

As discussed earlier, wireless sensor nodes are tiny, venerable electronic devices that are susceptible to damage; therefore, there is a high probability that some sensor nodes will be damaged during the deployment phase and will be unable to function as part of an operational wireless sensor network. Depending on their relative location, the loss of such nodes may lead to network partitioning by depriving a number of living nodes of the ability to deliver data to the sink. Likewise, unequal deployment of sensor nodes may also result in void regions, resulting in network partitioning. Similarly, in an operating wireless sensor network, many nodes may become unresponsive due to energy depletion or hardware failure, which might lead to network partitioning. In our proposed mechanism PIRCB, we consider two basic scenarios that may lead to network partitioning in a wireless sensor network:

- Partition creation due to void region created at sensor node deployment phase.
- Partition creation due to critical node failure during operational wireless sensor network.

To the best of our knowledge, no other partition identification method has taken into account the creation of partitions due to void regions that occur during the deployment phase of a wireless sensor network. It is well established that the partitions that arise during the initialization phase of a network are the most difficult to locate. Since the sink node has no clear knowledge of the status of these live nodes that are partitioned and the sensor nodes are unable to contact the sink in a conventional manner. Similarly during an operational network, the loss of a critically located key node can also lead to the formation of a partition. To deal with both partition creation scenarios , we propose a cooperative beam-based approach that allows the sensor nodes of the disconnected partitions to reach the connected part of the WSN that contains the sink node. The proposed partition detection and reconnectivity through cooperative beamforing mechanism is equally applicable to the above mentioned both scenarios. The proposed mechanism has three steps. First: partition detection, second beamforming at the reference node, and third: searching for connected relay nodes in order to establish connectivity to the sink node. These steps are explained in detail in the flowchart of the proposed mechanism in Figure.4.1.

4.3 Partition Detection

We consider a randomly deployed wireless sensor network consisting of N number of static sensor nodes represented as a vector

$$V = \{v_i \in V \text{where } i = \{1, 2, 3....N\}\} \tag{4.1}$$

over an area of interest. Once deployed all the operational sensor nodes are required to be connected to the sink node in order to communicate their collected data. Thus to ensure connectivity, all working nodes transmit their status information to the sink during the initialization phase of the network and wait a certain threshold amount of time (2RTT in our case) for the sink to acknowledge. If a sensor node cannot receive an acknowledgement from a sink node for a certain threshold amount of time, they consider themselves disconnected and update their status to be partitioned refereed as

$$V_p = \{v_i \in V | \text{Node who does not receive sink acknowledgement}\} \qquad (4.2)$$

The same procedure is used to detect and reconnect partitions created during the operation of the network due to the loss of a critically located nodes that serves as a path to the sink for a group of isolated nodes. The boundary nodes, where all messages accumulate and no further path is available, consider themselves as reference nodes and initiate the partition identification and connectivity restoration procedure. These reference nodes will select a cluster head on a single criterion of having high energy. The cluster head will gather information of all the sensor nodes belonging a to respective partition. Upon reception of this information it will create a vector for the ith partition P_i comprising of all the interconnected nodes that are unable to reach sink node and are therefore regarded as partitioned.

$$P_i = \{v_i | v_i \in V_p\} \qquad (4.3)$$

Following the construction of partition P_i, the cluster head will relay this information to the respective reference nodes to communicate it onward to the sink node through a cooperative beamforming.

4.4 Cooperative Beamforming

As mentioned earlier, a network would be said partitioned whenever the packets sent by the source are accumulated at a specific node on the path from the source to sink, and the said node is not able to get further acknowledgements from its receiving relay node ahead of it on the path towards the sink. Such node would then initialize cooperative beamforming procedure with the co-ordination of its neighbouring nodes. A node that initiates beamforming would be considered as the reference node, and its distance to the prospective receiving relay node would be taken as the x-axis of the frame of reference. To form a directional beam, the transmitted waves of the transmitting nodes should be received in phase at the receiver for their constructive summation. Nevertheless, the nodes of isolated cluster deployed in 2D planar area are not in an array form, and they do not have equal spacing among themselves. As an example, a disconnected cluster of three nodes N1, N2 and N3 is shown in Figure 4.2. These nodes aim to send their data to a receiving relay node R. Since three of the transmitting nodes know their physical locations, therefore they will easily arrange their phase adjustment according to their positions in order to form a co-operative beam towards any prospective receiver R. If these nodes start transmitting the same signal without phase adjustment, out of phase versions of these signals would be received at the receiver which would cause destructive interference. The obvious cause of this harmful interference is the path difference of the waves transmitted by the nodes located at different physical locations. However, if the to-be-transmitted waves are preprocessed at two nodes in accordance with the reference node before their transmission towards the receiver, the corresponding destructive interference can be converted into an in-phase summation resulting into an enhanced power at the receiver. Let d_{1R} , d_{2R} and d_{3R} be the paths taken by the waves transmitted from N1, N2, and N3 respectively as shown in Figure 4.2. Since N1 is taken as the reference node, it thus becomes the center of the x-y coordinate system. In order to form cooperative beam towards the receiver R , both of N2 and N3 should align their transmitting waves to that of N1 by using phase adjustment strategy; so that all waves should receive in-phase at R.

As the exact location of R is unknown, therefore the reference node N1 will have to visualize its location with the assumption that it may lie anywhere around it at a distance d_{1R} much larger than its transmission range. To ensure that signals transmitted by the disconnected cluster should reach R, the beam range should be equal or more than d_{1R}. However, the beam range strictly depends on the number of nodes that take part in the cooperative mechanism i.e.

$$d_{1R} \leq N d_0 \tag{4.4}$$

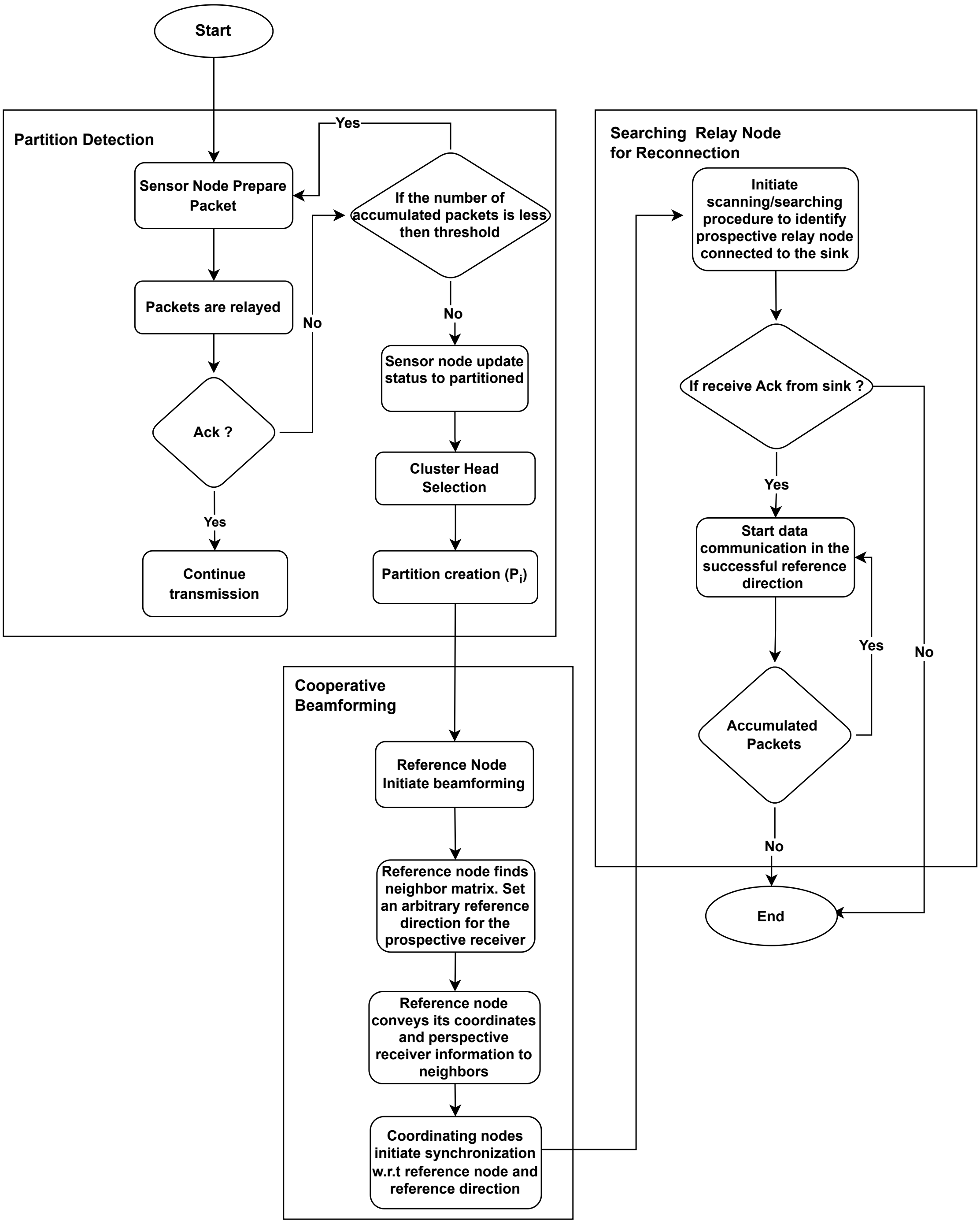

FIGURE 4.1: Flowchart of cooperative beamforming mechanism

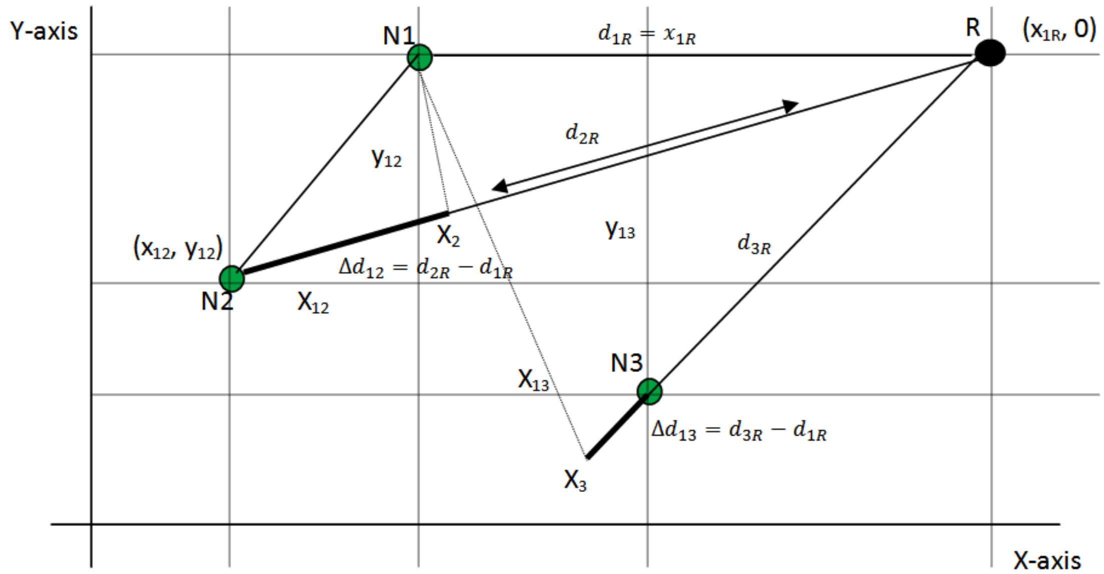

FIGURE 4.2: Working of the proposed Cooperative Beamforming Mechanism.

Where N is the number of total neighbouring nodes taking part in the collaboration including the reference node and d_0 is the range of transmission of a single node. It is worth mentioning that if d_{1R} is larger than Nd_0 even the cooperative beam would not be able to reach R and the disconnected cluster would never be able to send its data to the sink. It would thus remain disconnected from the main part of the network. Therefore, the initial distance between reference node N1 and the prospective unknown receiving relay node R would be set on the maximum beam range i.e.

$$d_{1R} = Nd_0 \tag{4.5}$$

Direction of the unknown receiving node from the reference node N1 is also crucial for directive beamforming. In the next section, we will present a detailed note on finding the direction of the receiving relay node and getting connected to it for data transmission. At the moment, we assume that R resides at any arbitrary direction from N1. As already discussed, the separation d_{1R} between N1 and R is set as the reference x-axis of the coordinate system, as shown in Figure 4.2. Considering the geometry drawn in Figure 4.2 and manipulating some basic calculations, the distance d_{2R} and d_{3R} can also be calculated as

$$d_{2R} = \sqrt{(x_{1R} - x_{12})^2 + y_{12}^2} \tag{4.6}$$

$$d_{3R} = \sqrt{(x_{1R} - x_{13})^2 + y_{13}^2} \tag{4.7}$$

where $x_{1R} = d_{1R}$ and $y_{1R} = 0$ because of N1 being the reference node set at origin with d_{1R} along the x-axis. The same mechanism can be utilized to find the distance between the receiving node and Nth transmitting node, taking part in the beamforming mechanism as

$$d_{NR} = \sqrt{(x_{1R} - x_{1N})^2 + y_{1N}^2} \tag{4.8}$$

From the geometry shown in Figure 4.2, Δd_{12} and Δd_{13} are the differences in the distances traveled by the waves transmitted from N2 and N3 with reference to N1, respectively.

$$\Delta d_{12} = d_{2R} - d_{1R} \tag{4.9}$$

$$\Delta d_{13} = d_{3R} - d_{1R} \tag{4.10}$$

In general, Δd_{1N} can be calculated for any node N as

$$\Delta d_{1N} = d_{NR} - d_{1R} \tag{4.11}$$

This distance disagreement Δd_{1N} covered by the wave transmitted from the node N as compared to the one transmitted from reference node N1 will cause a phase difference at the receiver. Magnitude of this phase difference can be calculated as

$$\Delta \phi_{1N} = 2\pi f_c(\Delta d_{1N}/c) \tag{4.12}$$

Where f_c is the operating frequency and c is the velocity of light. If we can implement a pre-processing technique that calculates the phase difference at each node with respect to a reference node $N1$ prior to transmission, it would enable the receiver to receive all waves in-phase, regardless of their path differences. Essentially, this means that all nodes participating in the cooperative mechanism would have the capability to form a powerful beam, ideally amplifying the power by a factor of N^2 compared to a single node. Consequently, this enhanced transmit diversity empowers them to transmit data over significantly longer ranges, enabling improved communication coverage and reach. By harnessing this advanced transmit diversity, wireless sensor networks can extend their capabilities and facilitate the seamless transmission of data across broader distances. as shown in Figure 4.3. This will also improve the bit error rate of the communication link. In this

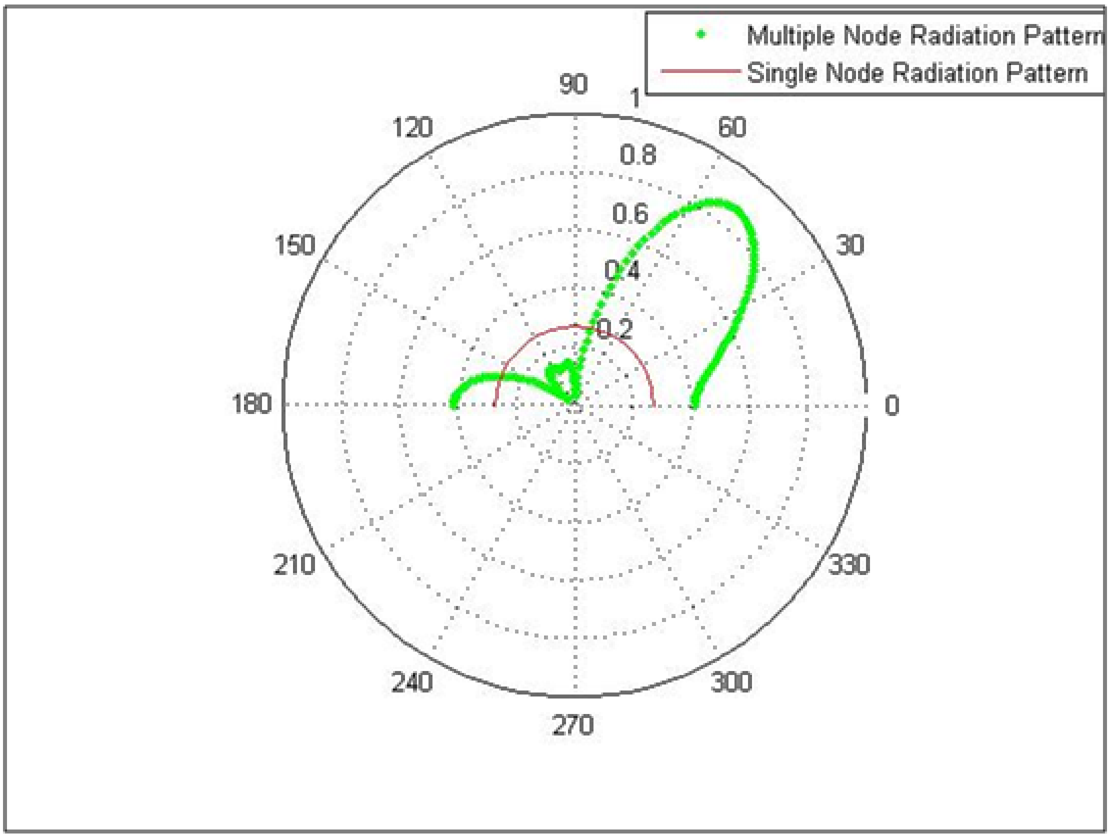

Figure 4.3: Single Node Radiation Pattern versus Collaborative Beam Pattern.

way, the nodes in the disconnected cluster will become able to form a cooperative beam towards any closest node attached to the main part of the network.

4.5 Searching Nearby Connected Relay Node for Reconnectivity

After forming cooperative beam, the nodes in the disconnected cluster would still not be able to connect to the main part of the network, if they do not know the direction in which the nearest relay node (already connected to the main part) resides. We present the following mechanism to address this problem.

Algorithm 4: Cooperative beamforming based scanning procedure for searching any alive node connected to the main part of the network.

$\Delta\theta \longleftarrow 0.5°$; $\theta \longleftarrow 0$; `/* Initialize beam direction */`
BF_flag $\longleftarrow 1$; `/* Set the beam flag to true */`
repeat
  Time_slot$\longleftarrow$ `getSystemTime()` ; `/* set the time slot corresponding to the beam direction */`
  Hello_msg$\longleftarrow v_i + p_i + BF_flag + Time_slot$; `/* Beacon message */`
  `transmit(`*Hello_msg*`)`;
  $\theta \longleftarrow \theta + \Delta\theta$;
until $\theta \leq 360$;
Ack_msg$\longleftarrow$ `listen(`*Ack*`)` ; `/* Listen for acknowledgment from the sink */`
if *Ack_msg=Null* **then**
  quit() ; `/* Failure: No alive relay node in range */`
else
  Success_angle $\longleftarrow$ `extract(`*Ack*`)` ; `/* Success: alive node in range at angel` θ `*/`
  `transmit(`*Data_packet*`)` ; `/* Start data transmission */`
end

When the nodes of the disconnected cluster become able to generate an intense beam which covers a longer distance, they start scanning the whole area around them for the possible existence of an alive relay node which may connect them to the main part of the network. The sequence of steps needed to perform this function is presented in Algorithm 4. In this algorithm, the reference node in the disconnected cluster transmits beacon/hello messages in narrow beams in all directions in a cyclical manner(in a clockwise or anticlockwise direction).

Algorithm 5: Procedure followed by the sink on successful reception of beacon message generated through cooperative beamforming.

```
Max_RSSI ⟵ 0; timeThreshold ⟵ 2 × RTT ;          /* Set the time
 threshold to twice the length of round trip time */
for i=0 to timeThreshold do
   Rcv_msg=listen(Hello_msg) ;       /* Wait for beacon message */
   if BF_flag=true ;                     /* Check the beam flag */
    then
      Rcv_RSSI ⟵ extract(Rcv_msg);   /* extract RSSI information
       from the received message */
      if Rcv_RSSI > Max_RSSI ;          /* Look for maximum RSSI */
       then
         Max_RSSI ⟵ Rcv_RSSI ;
      end
   end
   Max_RSSI_msg ⟵ Rcv_msg;
   Time_slot ⟵ extract(Max_RSSI_msg);   /* Extract the time slot
    corresponding to the beam direction from the message
    received with maximum signal strength at the relay node */
end
transmit(Ack Times_slot);    /* Send acknowledgement with time slot
 corresponding to the successful beam angle */
```

The beacon/hello message $Hello\_msg = (v_i, P_i, BF\_flag, Time\_slot)$ contains three types of information: one about the disconnected partition P_i and its respective identification and location of reference node v_i. This allows the sink to notify the application of the redeployment of relay nodes while maintaining connectivity during this time-consuming transition period through cooperative beamforming. The second piece of information states that the beamforming flag is enabled, and the third is about the time slot at which the beam is formed in the respective direction. The flag will tell the receiving node about the special circumstances of the partitioning due to which the message was generated by cooperative beamforming.

If any relay node, connected to the main network, receives any of these beams containing specific information (BF_flag), it will measure the received signal strength (RSSI). The RSSI is transmitted to sink along with the received message.

The procedure followed by the sink on the successful reception of the message generated through cooperative beamforming is explained in Algorithm 5 as the sink may receive many messages arriving via multiple relay nodes with different beam power level information (RSSI). It selects the node with the highest RSSI as a valid relaying agent for communication with the disconnected cluster. The time slot information ($Time\_slot$) contained in the beacon message having highest RSSI will identify the successful beam direction at the source node. The sink will acknowledge the receipt of the highest RSSI beacon message by sending acknowledgment directly to the disconnected cluster along with the time slot ($Time\_slot$). The time slot will help the group to set the respective direction corresponding to the strong beam as the correct direction for future communication. The reference node in the group, also working as the cluster head asks the other nodes participating in the cooperative beamforming mechanism to lock their phase adjustment for the auspicious time slot for future communication with the central part of the network. The sink may receive the cooperative beam directly without getting hopped by any relay node. However, the same procedure of acknowledgment will be followed. Upon successful completion of this process, the isolated cluster will be able to send the important time-critical packets accumulated at different reference nodes to the sink.

4.6 Results and Discussion

Table 4.1: Packets allocated to different nodes

Number of Source Nodes	Number of Packets Generated Per Source
19	$Packets > 50$
14	$40 < Packets < 50$
12	$30 < Packets < 40$
5	$Packets < 30$

The proposed scheme is evaluated for received power enhancement to ensure network connectivity by partition healing using cooperative beamforming. Several rounds of simulation are performed, using MATLAB R2021a® for each scenario.

The data points are taken as an average of 10 simulation rounds to ensure the result's precision. Each simulation is run over the sensor nodes distributed uniformly covering an area that varies from 500 m^2 to 4000 m^2 keeping the number of nodes and their communication range constant. By examining the scheme's efficacy under different node densities, we can gain insights into its performance characteristics. This analysis provides valuable information for evaluating the scheme's suitability and effectiveness in diverse deployment scenarios.

For better understanding besides the detailed experiment setup, we present a sample experimental setup of the simulation field of $100m * 100m$ comprising 50 sensor nodes.

The following general assumptions are made for the simulation.

1. Initially, all the sensor nodes have the same power.
2. Nodes are distributed randomly in 2D plane.
3. Nodes are static, i.e. they can not move.
4. The collaborating nodes are assumed to be in line of sight with the reference node, where delays are calculated through physical distances among transmitting nodes.
5. The transmitting nodes are equipped with omnidirectional antennas.
6. The communication channel between the transmitting nodes and the sink is linear with time-invariant response.
7. Nodes are aware of their respective location through inbound GPS devices.

The initial deployment is random, and each node is assumed to have a constant communication range and bit rate but varying packet rate, as shown in Table 4.1.

The change in topologies with variable packet rates forces some nodes of the network to deplete their energies faster than the others, which causes bottleneck in the network if such nodes are located at critical locations, such as node labeled 39 as shown in Figure 4.4. This bottleneck eventually leads to network partitioning. In other words, the elimination of the node 39 causes the cluster of nodes 4,15 and 40 to partition and disconnect from the central part of the network. A cooperative beamforming mechanism is thus executed which results in the generation of high power beam as shown in Figure 4.5

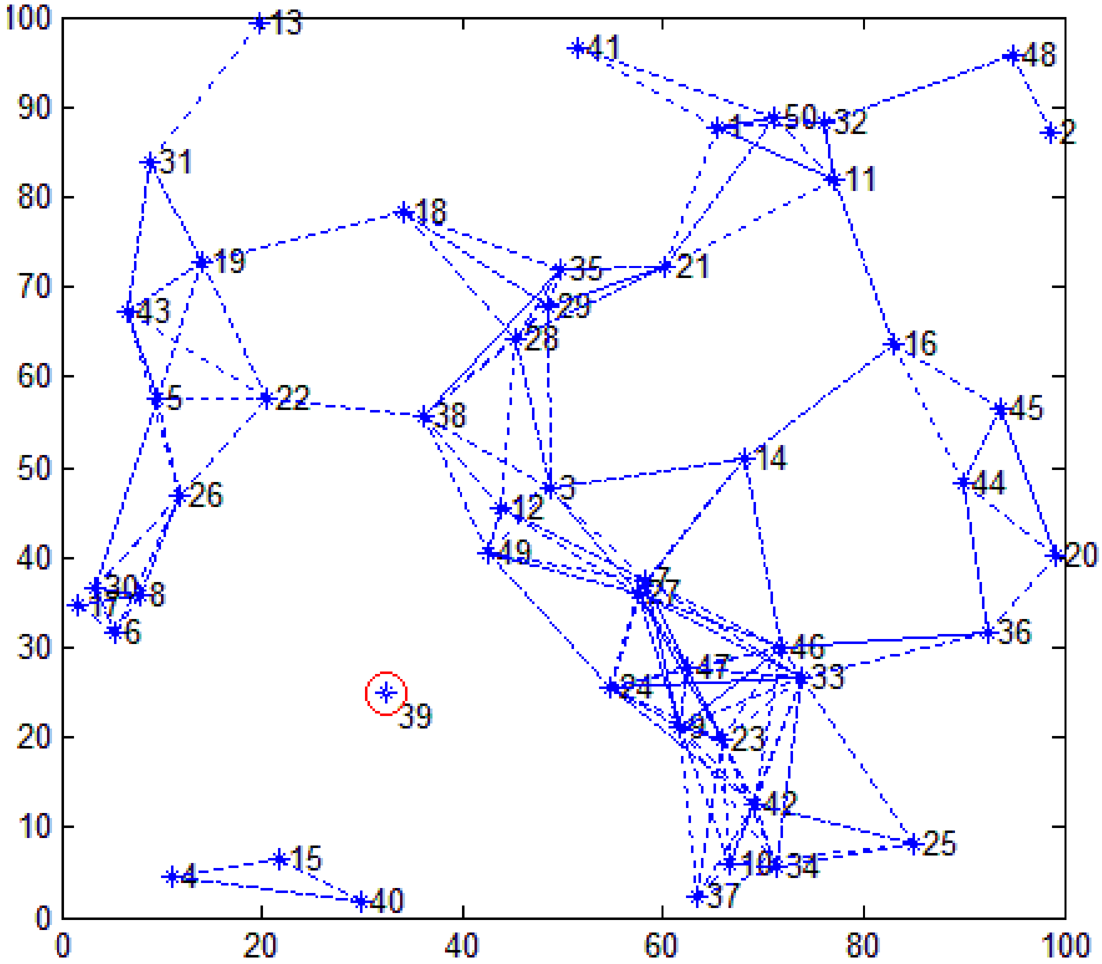

FIGURE 4.4: The wireless sensor network with disconnected Partition.

Figure 4.6, depicts the cooperative power analysis of the scenarios, when beamforming is available or not available, in terms of the distance between the transmitter and the relay node. The result shows an exponential decrease in received power of the signals transmitted individually by node 15 with increasing distance, and eventually, these signals become undetectable at the closest relay node numbered 49. However, after the formation of cooperative beam, the collaborative effort of all three nodes in the disconnected cluster enables them to reach the nearby relay

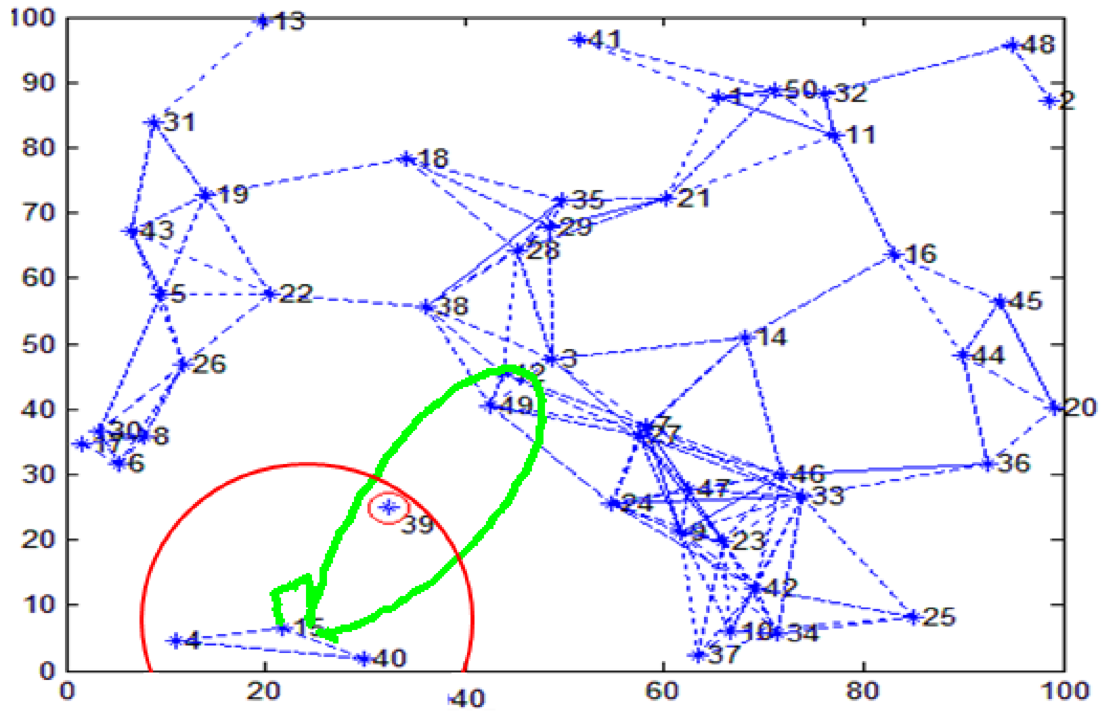

FIGURE 4.5: Connecting isolated cluster through proposed cooperative beamforming mechanism.

node and thus to resume the communication with it. This enables the resumption of data transmission and facilitates the restoration of network connectivity. highlighting the potential of cooperative strategies in mitigating connectivity disruptions. For detail evaluation, we take a comprehensive simulation setup comprising 300 nodes with varying simulation area as mentioned in Table 4.2.

The results are obtained with varying node densities to achieve different scenarios of partitioning. We can only execute our scheme of partition healing when a reference node (the one that initiate beamforming) has sufficient number of neighbours. A significant reduction in the number of disconnected isolated partitions can be observed in Figure 4.7

as an average of ten rounds each. The trend of the number of disconnected partitions with and without using the beamforming approach can be observed. It can be noticed that at the simulation area between $2000m^2$ to $2500m^2$, the difference between the curves is quite significant, which means that beamforming mechanism is successful enough to heal a good number of partitions when partitions are

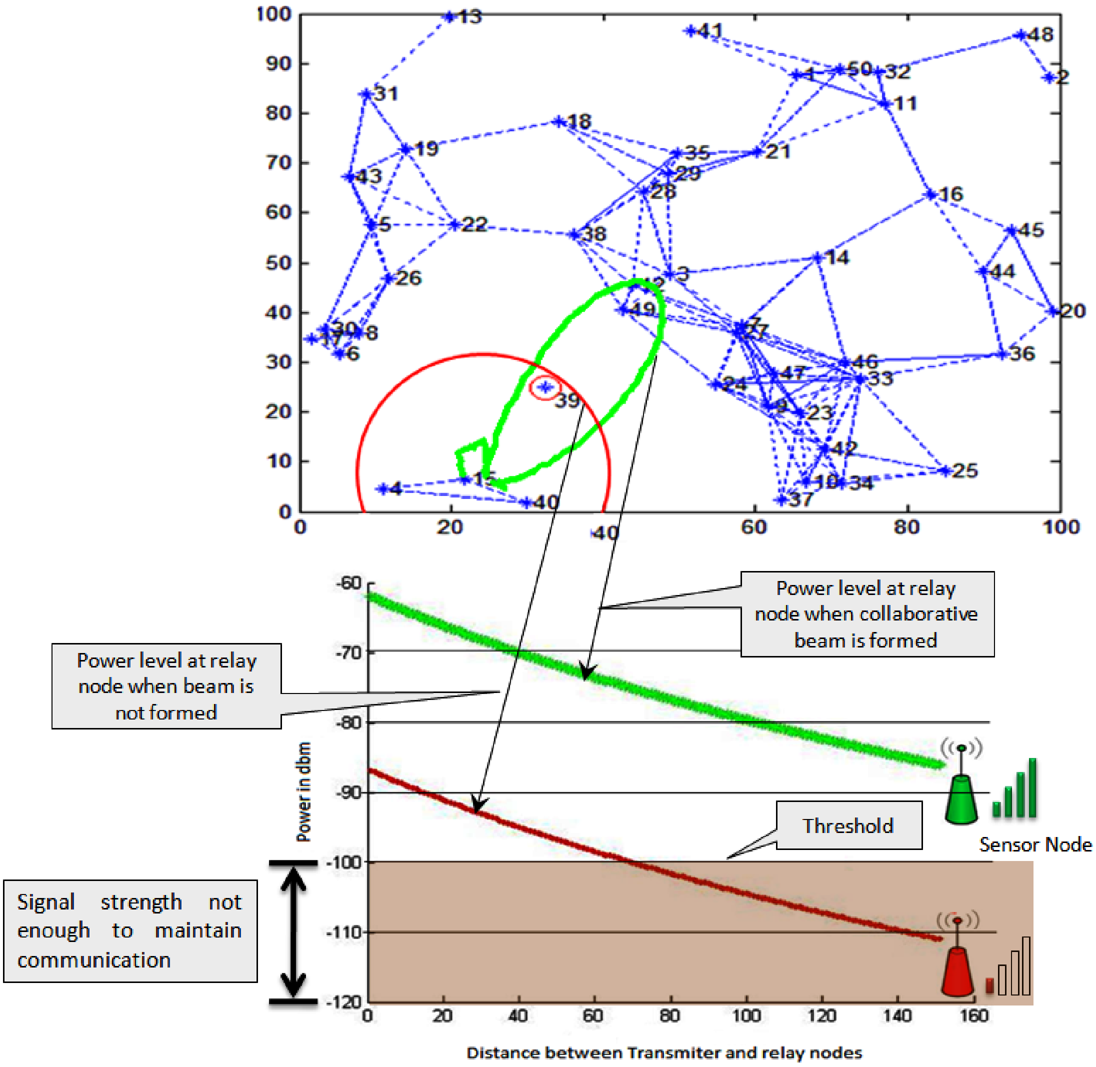

FIGURE 4.6: Power analysis of the cooperative beamforming as compared to individual node transmission in the disconnected cluster.

comprises of large number of nodes. This is because the efficiency of our proposed approach relies on the number of participating nodes in cooperative beamforming. Similarly, as we increase the simulation area, the difference between the curves decreases and they appear to converge. This is because the average node density tends to reduce with increase in simulation area resulting in large number of single- node partitions which cannot be handled by our proposed scheme. Figure 4.8 shows a percentage improvement in partition healing. The curve shows that we can achieve up to 70 % reduction in network partitioning if isolated partitions have a sufficient number of nodes to participate in cooperative beamforming. In Figure 4.9, we can observe that by incorporating cooperative beamforming we can

Table 4.2: Simulation Parameters for comprehensive simulation setup

Simulation Parameter	Value
Simulation tool	MATLAB R2021a ®
Routing Protocol	GRACE
Area	500 m^2 to 4000 m^2
Number of nodes	300
Communication range	50 m
Energy Consumption in T_x	14.87 mJ
Energy Consumption in R_x	15.39 mJ
Energy Consumption in Listening	3.5 mJ
Data Racket Lenght	1260 Byles
Communication Range	100 m
Node Distribution	Uniform Random
Number of Simulation rounds	10-15
Data Rate	256 kbps

significantly improve the network lifetime. In addition, we also perform a comparative analysis of the proposed technique PIRCB, with two other recent techniques DPDRU [38] and IDCRWSN [88]. The first technique, DPDRU, assumes a homogenous static wireless sensor network topology and employs a reactive neighborhood awareness-based partition identification technique for recovery through static node redeployment using unmanned aerial vehicles. The second technique, IDCRWSN, uses an intelligent on-demand connectivity restoration technique for wireless sensor networks and proactively responds to node failures by relocating a redundant relay node to replace the failed node. These two baseline techniques are representative of the most widely used partition recovery techniques, i.e., reconnectivity through additional static node redeployment and through relocation of mobility aware existing sensor nodes to establish connectivity in a timely manner. Thus, we believe that the chosen baseline methods provide a fair comparison and are representative of the state-of-the-art in partition detection and recovery.

Figure 4.10 describes the network efficiency in terms of the number of successfully

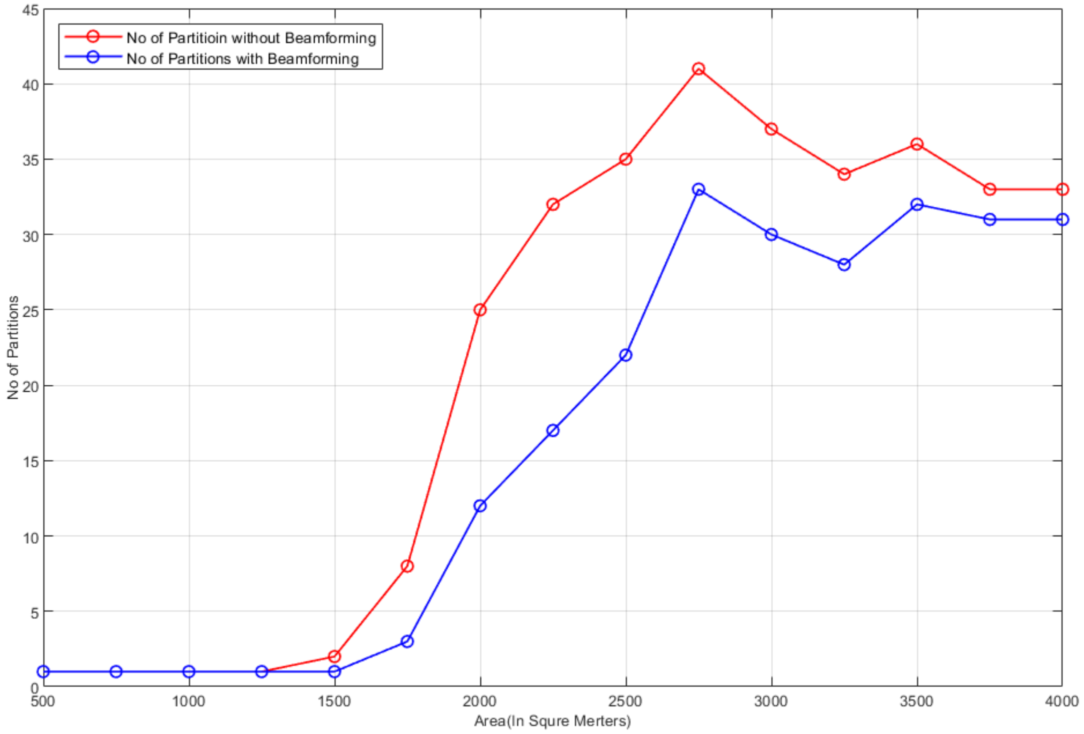

FIGURE 4.7: Partition Development/creation with and without using Beamforming

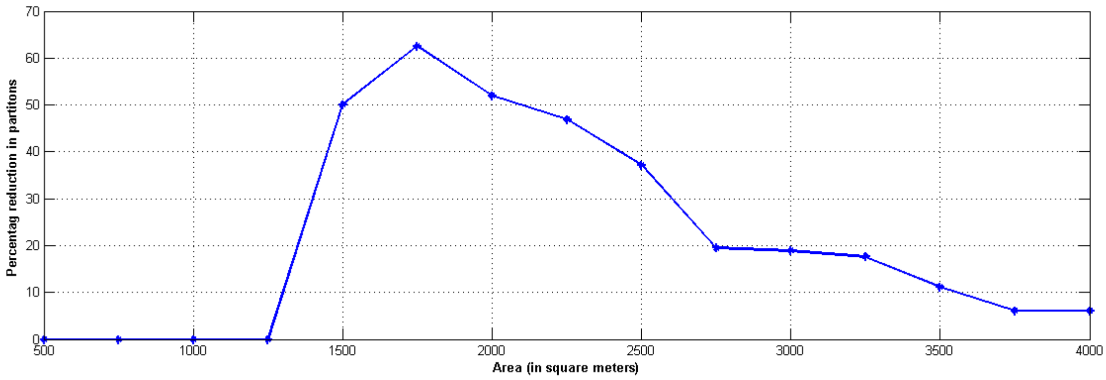

FIGURE 4.8: Percentage reduction in partitions through beamforming as a healing process

received packets with respect to the alive nodes. This shows that the network remains stable over a longer period of time when cooperative beamforming based partition healing is incorporated. In other words, by maintaining network connectivity, we can achieve higher throughput and avoid critical data loss. Since our

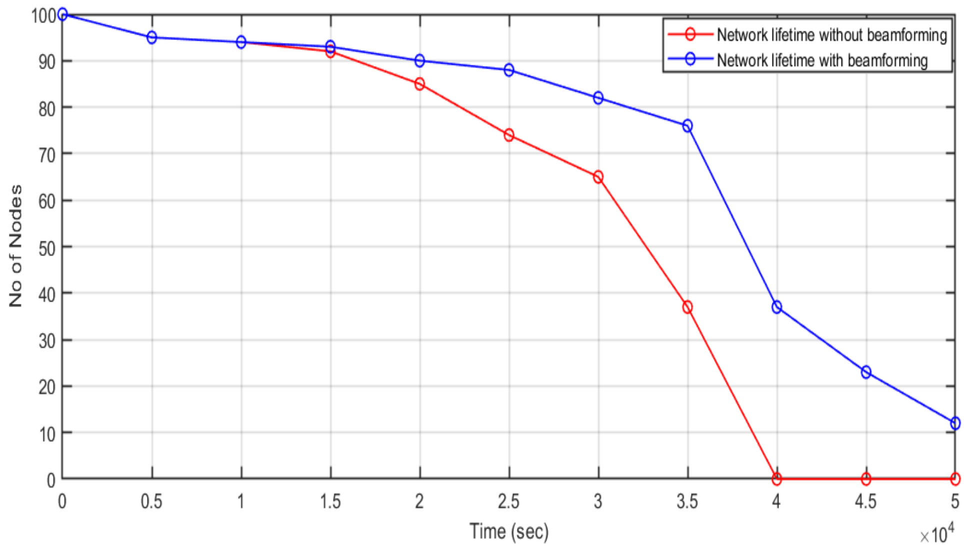

FIGURE 4.9: Network lifetime with and without using Beamforming

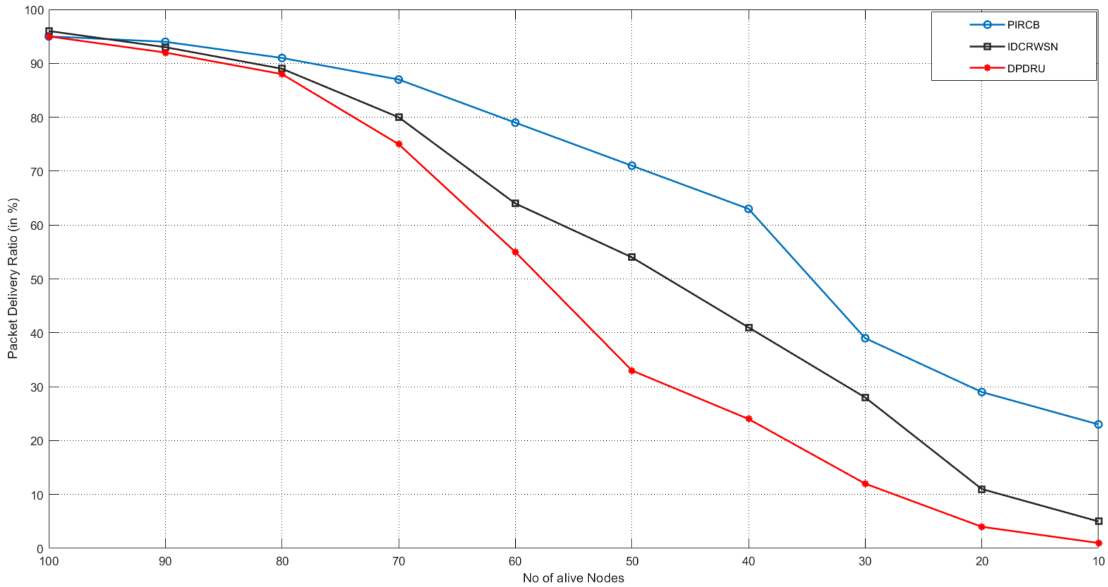

FIGURE 4.10: Packet delivery ratio comparison

proposed technique does not consume any additional resources and the partition identification and healing is performed in near real-time; Therefore the packet loss ratio is very low compared to the other two approaches. In contrast to DPDRU and IDCRWSN approaches, which rely on the deployment of additional nodes to resolve network partitioning, our proposed technique, PIRCB, offers distinct advantages. One of the key differentiators is the reduced time and minimized

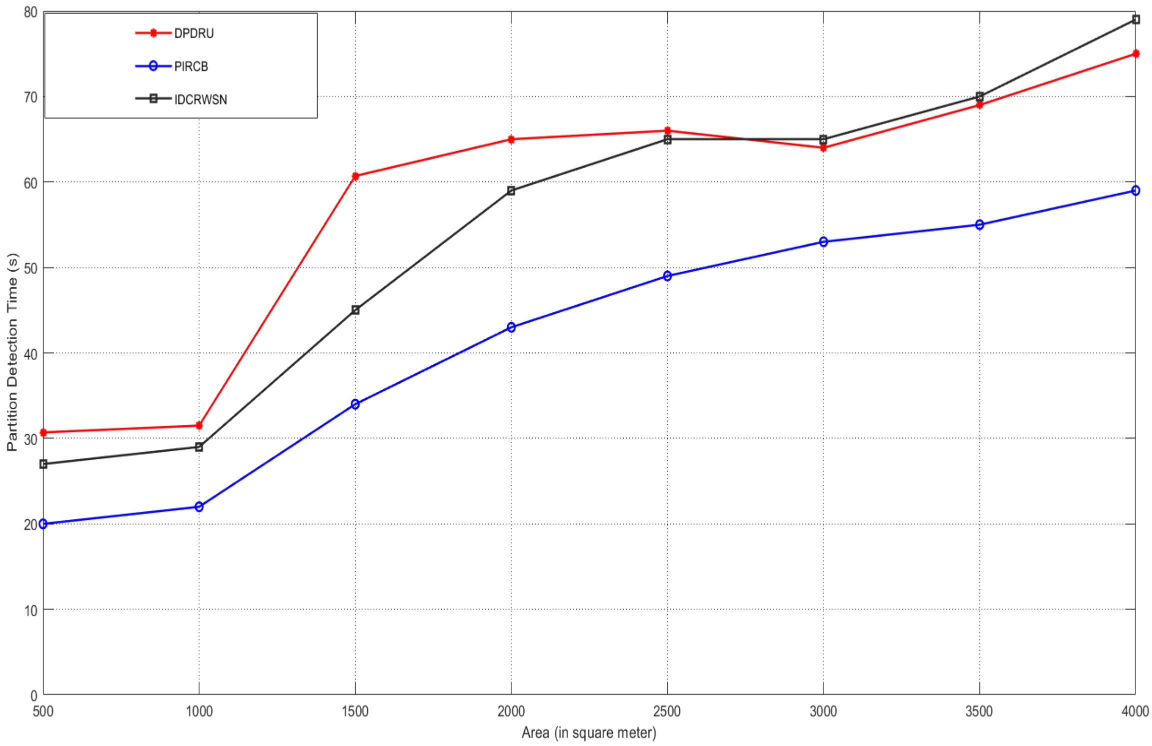

FIGURE 4.11: Partition detection time comparison

packet loss associated with partition healing. This can be observed in Figure 4.11, which presents a comparative analysis of partition detection time. As depicted in the result, our PIRCB technique outperforms both DPDRU and IDCRWSN in terms of partition detection time. This superiority can be attributed to the distributed nature of PIRCB, which eliminates the need for additional messaging among nodes. By avoiding the extra communication overhead and associated delays, PIRCB significantly enhances the efficiency and effectiveness of partition detection and resolution. These advantages make PIRCB an appealing choice for addressing network partitioning in wireless sensor networks. By minimizing the time required for partition detection and avoiding excessive packet loss, PIRCB offers improved network performance and contributes to the overall reliability and robustness of the system.

4.7 Conclusion

In this chapter, we have proposed a cooperative beamforming-based approach to retrieve valuable and time-critical data from the nodes trapped in the disconnected cluster of a wireless sensor network. In the proposed scheme, we have addressed both scenarios of the partitions created at the time of deployment as well as those created during the operation of the network due to node failures. In this scheme, Sensor nodes trapped in isolated partitions cooperate to form a directional beam that significantly increases their overall communication range to reach out to a remote relay node connected to the main part of the network. In addition, this mechanism increases the fault tolerance ability of a WSN without requiring additional nodes, minimizing the loss of its critical data in the event of network partitioning. This provides a robust solution for data-critical wireless applications.

Chapter 5

Energy Visualization and Re-Deployment of Nodes in Wireless Sensor Networks

5.1 Introduction

A wireless sensor network is an ad-hoc network, consisting of sensor nodes that are interconnected. Data is forwarded on a hop-by-hop basis, i.e., each node forwards incoming data to the destination, as described in the previous chapters. This entire process consumes energy, so the total energy of the nodes and the network gradually decreases. The decrease in the energy of the node(s) can lead to two scenarios: individual nodes are disconnected and disconnected or void regions (network partitions) are created. we consider nodes disconnected when they have so little energy that they cannot process or send data. whereas disconnected regions/partitions occur when nodes that connect to the main part of the network with the sink are disconnected. In each of the above scenarios, the disconnected nodes must be replaced, either by deploying additional sensor nodes or by moving an existing sensor node to the region in question. However, most of the contemporary systems make the decision of re-deployment after nodes have died out or empty regions have formed. Thus, this reactive approach can take more time. Moreover, they operate autonomously, without human judgment, and mostly based on probabilistic models. Therefore, current systems are time consuming, lack precision and not suitable for real-time WSN applications . As the energy of nodes in the network gradually decreases and nodes die out at a certain point, node extinction is a gradual process that occurs over a certain time period. We can use this information to visualize the energy status of sensor nodes in such a way that it enables a network manager to identify critical sensor nodes and predict a possible network partitioning. Using information visualization and visual analysis, we can enable the network manager to recognize the expected and unexpected semantic information contained in the data, and enable the manager to combine automated methods with expert intuition to identify complex patterns of abnormal network activity. Information visualization is a method that has been used for a long time to make information clear and easy to understand by using graphical means. The main goal of visualization is to make visual representations of information that people can interact with and use their visual perception and problem-solving skills to get the required information for decision-making. For this reason, more and more researchers agree that visualization is a useful new tool for visual data mining [26]

Information visualization is used in a variety of applications, including network data visualization . Visualizing network traffic is probably one of the first steps in understanding, evaluating, and finding relevant information in the large amounts of data in a network. Numerous visualization tools, including scatter plots, colour maps, etc., have been developed to assist network analysts in their tasks [115]. In the WSN field, several network visualization tools have been introduced to graphically monitor real-world or simulated sensor deployments. Existing technologies such as PROVIZ [116] , Spyglass [117], NetTopo [118] and Surge [28] display network activity between sensor nodes and provide the user with real-time information about the network topology and collected sensor data to enable real-time troubleshooting of a deployed sensor network [119]. Although the aforementioned standard data visualization tools, as they stand, add a new dimension to network traffic monitoring, they lack the specialized visualization application required to use information about the energy status of sensor nodes to manage network partitioning and optimize redeployment to ensure connectivity.

To solve the problem, we have proposed human assisted network visualization to optimize network connectivity.The proposed visualization technique will enable a network manager to: (i) identify potential nodes that are about to die out at a certain time; (ii) identify cuts/partitions in the WSN; and (iii) detect energy holes in the WSN. (iv) Recommend redeployment at certain locations to avoid disconnectivity.

The rest of the chapter is organized as follows: section 5.2 describes the proposed idea. Section 5.3 presents the architecture of the proposed system. Section 5.4 elaborates the visualization technique. The results and evaluations have been demonstrated in section 5.5 and 5.6 respectively.

5.2 Proposed Idea

The goal of this study is to use energy visualization to find out which sensor nodes are important and whose failure could cause network partitioning. In addition, to assist the network manager in optimizing redeployment by determining the best time and location to deploy more sensor nodes in order to keep the network connected. If one or more sensor nodes on key communication paths run out of power and are no longer able to participate in active network functions, this could lead to cuts that split the network into separate parts causing network partition as shown in Figure 5.1 and Figure 5.2. Since sensor node energy depletion is a gradual process in which nodes lose energy over time, we can track energy depletion to predict node failure before it occurs. This proactive approach can help us solve the problem before it occurs. In other words, we can say that this approach can help us avoid the problem before it occurs. In wireless sensor networks, node failure cannot be avoided in a hostile environment. Therefore, we can use this approach to avoid undesirable circumstances, such as the occurrence

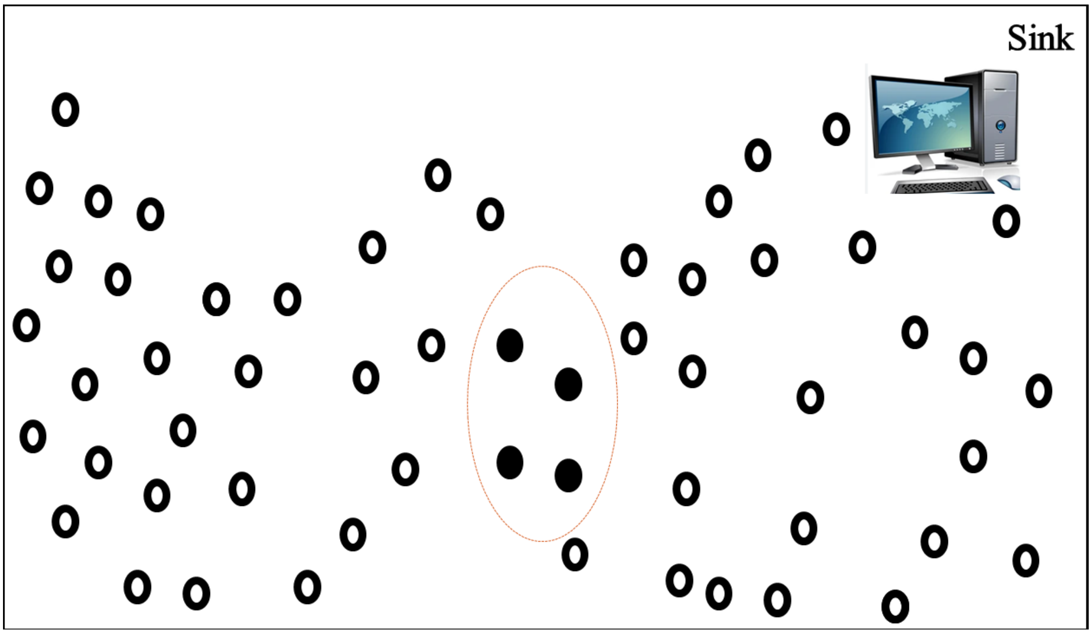

FIGURE 5.1: Cuts in wireless sensor network

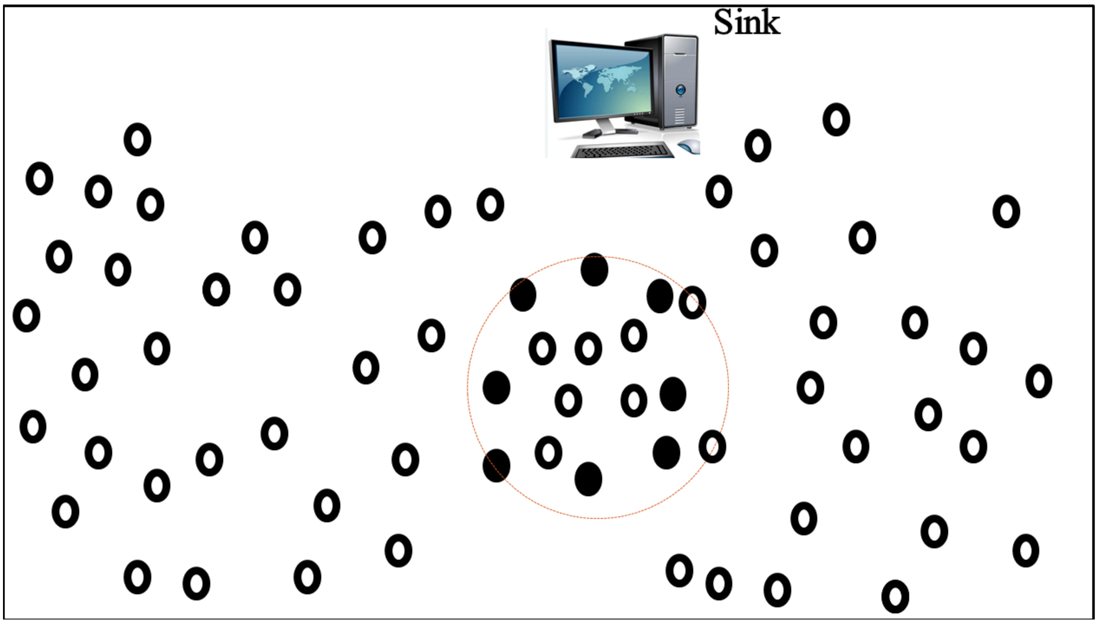

FIGURE 5.2: Partition in Wireless Sensor Network

of partitions or interruptions due to node failures at key locations. The proposed system employs a visualization technique in WSNs to visualize the energy of the active sensor nodes to obtain a visual image of the entire network and its energy status information. We assigned certain colors to indicate the energy status of the nodes. For example, the green color indicates that the node has the maximum energy, while the black color indicates that the node has the minimum energy state. We set certain threshold values for the energy state, and the colors are based on those thresholds. To keep the network connected, we must redistribute the sensor nodes, but the redistribution must be optimal in order to save cost and labor while providing the best possible results. Our method helps managers and the deployment team figure out exactly where expected cuts will happen and where nodes need to be re-deployed to avoid network partitioning, i.e., how many more nodes need to be put in place in the exact area. Unlike the traditional techniques, our technique is helpful in pointing out the redeployment region even if the nodes are not dead yet. However, the trend shows that they will die soon because the energy of that region or node is low enough that the node(s) will fail soon, which may disconnect the region from the rest of the network. We used the dataset of a well-known routing algorithm, GRACE [120], which is realistic and well tested on the testbed of Java Sun Spot sensors.

5.3 System Architecture

The proposed methodology is based on architecture with some interconnected modules, namely sensor node, deployment, sensor network; identify partition, energy mapper, node status, Redeployment, and the human as shown in Figure 5.3. We will briefly explain all of them in the coming lines.

5.3.1 Sensor Node

The sensor node serves as a fundamental component for detecting and monitoring a wide range of phenomena, encompassing areas such as temperature, light, humidity, motion, and numerous other aspects pertinent to various domains. However, it is important to note that these sensor nodes are subject to limitations in terms of available resources, including memory, energy, computational capability, and bandwidth. For the purposes of this particular study, the utilized sensor operated at a frequency of 2.4GHz and facilitated data transmission at a rate of 50Kbps.

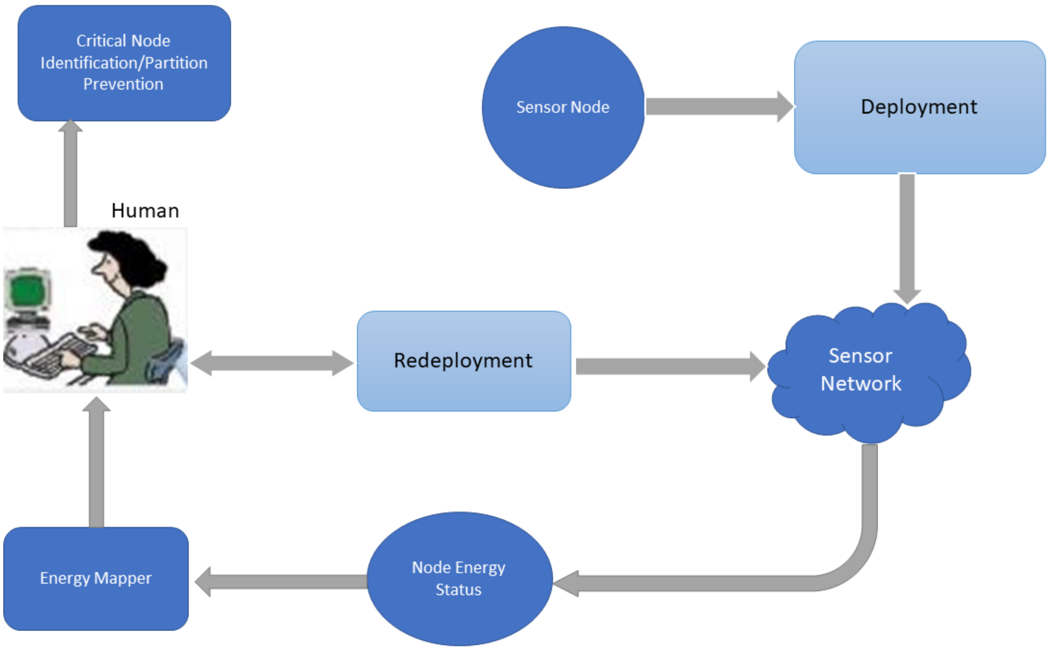

FIGURE 5.3: System Architecture

5.3.2 Deployment

In this module, the sensor nodes are first placed at suitable locations. It is not always possible to place every node in the correct location, especially if it is not done manually. In most cases, placement is not done manually, so dense placement is recommended to avoid network partitions or at least minimize the probability of network dis-connectivity. In our case, the initial deployment is random, dense, and as accurate and optimal as possible. The purpose of such a deployment is to prolong connectivity and keep the network healthy to the maximum possible extent. Our deployment area is 1000x1000 meters, with a random and dense deployment by keeping in mind that all nodes and the sink are stationary.

5.3.3 Sensor Network

When sensor nodes are deployed in a certain area of interest and start communicating, it is called a sensor network. Since it consists of sensor nodes, it inherits the constraints of sensor nodes and also has its own problems, such as channel impairments, connectivity problems, routing problems, power management problems, congestion, collisions, and security, etc. Our sensor network consists of sensor nodes and sinks. The routing protocol used in this network is GRACE (GRAdient Cost Establishment) [120] which is one of the better energy-efficient routing protocols with an event-driven data generation model at the source node. Our sensor network is homogenous in nature.

5.3.4 Node Status

This is another module of our architecture that is fed by the sensor network. The node status module defines and explains the status of each node, such as how much energy the node has at the moment, and provides other information required by the energy mapper module.

5.3.5 Energy Mapper

This is the most important module, which serves as the technique foundation. This module gets the energy level for each node received at the sink and maps it accordingly. For the purpose of mapping, we make use of the visualization tool TIBCO Spotfire [121], generate a scatter plot, and assign a color to each node in accordance with the energy level that it represents. The color green will show the maximum amount of energy, while the color black will indicate the smallest amount of energy. There are five unique categories, and each one is denoted by a color in addition to a percentage difference between 0 and 100 that ranges from 0 to 20%.

5.3.6 Identifying Potential Partition

This module is crucial for deciding at what point connectivity is going to be lost. In our case, we only tell the network manager the energy level and do not force a decision on it. This module is coupled with the human module, or, in other words, the network manager and his team. This module identifies critical nodes with respect to their location and energy levels and determine whether the loss of these nodes will lead to network partitioning.

5.3.7 Redeployment

The scope of the redeployment module is to redeploy the new sensor nodes that replace the failed nodes. This redeployment should be done at suitable and optimal positions. The main objective of the redeployment is to overcome the problem of disconnectivity and eliminate cuts, holes, which leads to network disconnectivity. This module is the most important part of the whole architecture. It determines whether the visualization achieved its goal by determining the optimal location for sensor redeployment to optimize connectivity. Moreover, there is a close cooperation between the generation of results by the visualization and the enforcement of these results by humans, so the redeployment team must be agile and careful.

5.3.8 The Human

The network manager and his team are collectively referred to as the human module. The main task of this module is to carefully monitor and check the visualization of the network against the incoming energy data from the sensor nodes before taking re-deployment decisions. After that, it is up to this team to decide whether more sensors should be placed somewhere or not. An intriguing aspect of our proposed technique is its flexibility and adaptability. It does not impose resistance if the network manager deems it necessary to deploy more sensor nodes, even if the visualization results indicate sufficient energy in that particular region. The proposed algorithm aims to provide the network manager with precise and up-to-date information regarding the current state of the network, without enforcing any predetermined decisions. The final determination is left entirely to the discretion of the network manager, who possesses the autonomy to make informed choices based on the given information and their expertise.

5.4 Visualization

For smooth and efficient redeployment of additional sensor nodes to ensure network connectivity in the sensor network, we have developed an innovative visualization technique that addresses the following issues: 1) The energy level (energy strength) of each node must be represented; 2) holes in the network must be identified, and 3) cuts/partitions in the network must be identified. We used color coding for different energy levels of sensor nodes. In our system, the energy of each node is communicated to the sink, which is then used by our Energy Mapper module for visualization. This visualization displays the sensor nodes according to their energy level. For example, the green color represents the sensor node with the highest energy level. The black color indicates the "died out" or "about to die-out" nodes. The nodes that have neither the maximum nor the minimum energy are represented with the following colors: blue, yellow, and red, depending on their energy level. The energy is given in percentages from 0% - 100%, where 0% represents the minimum energy level and 100% represents the maximum energy level. We interviewed ten experts in the field to find out how to match energy levels with the right color codes. Based on their feedback, we assigned color codes to each energy level, as shown in the table 5.1. Nodes with an energy level between 80 and 100 percent, for example, are considered "green" in the network. The nodes with an energy level lower than 20% are considered black nodes. However, this color coding scheme is generic and can be customized by the administrator.

5.4.1 Dataset

To obtain the dataset for energy visualization, we simulate one of the most stable and well-known routing protocols GRACE [120], a routing protocol that maintains the energy state information of sensor nodes to achieve energy efficient routing in a WSN. There are more than 4000 readings for 50 nodes in a simulation area of 1000mx1000m, as shown in Figure 5.4. These results were obtained using MATLAB simulation tool. To obtain accurate results, a series of simulation rounds is

TABLE 5.1: Color Coding Scheme for Energy Levels of Sensor Nodes.

Assigned Color	Energy Level
Green	80% to 100%
Blue	60% to 80%
Yellow	40% to 60%
Red	20% to 40%
Black	0% to 20%

performed. The table 5.2 explains various parameters used to create the dataset for this visualization technique.

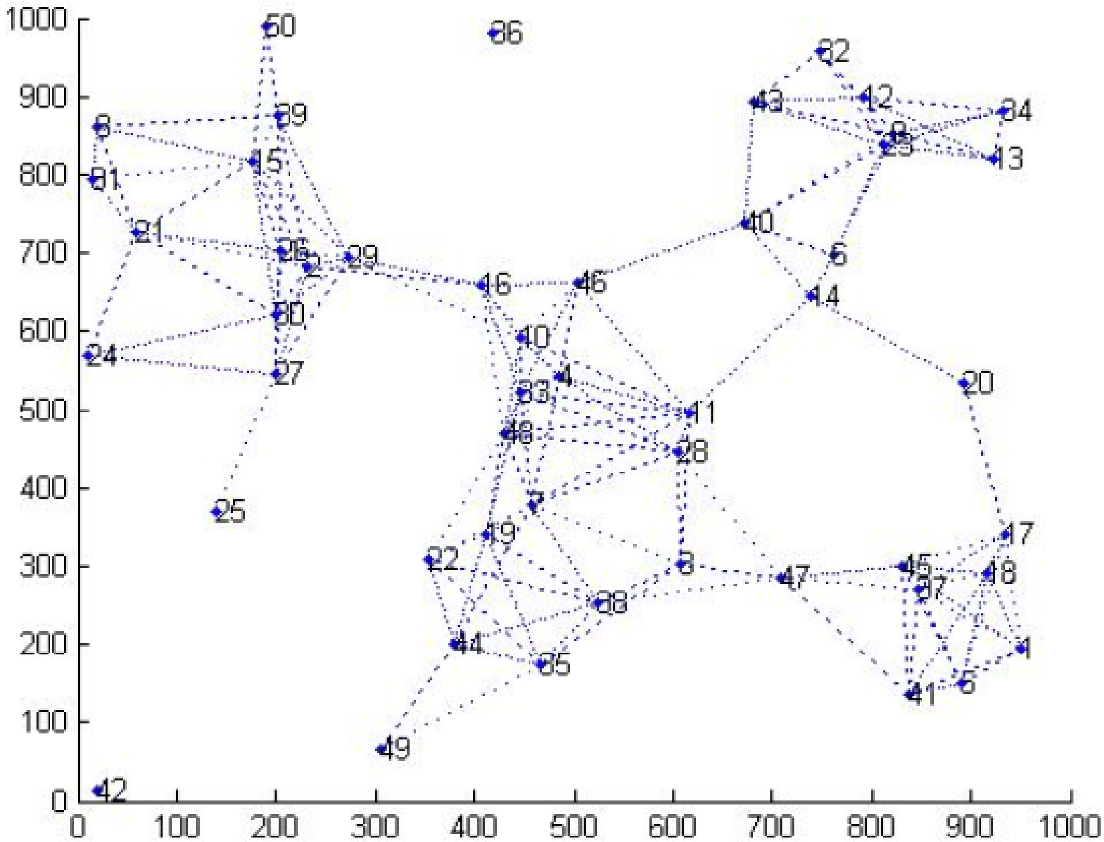

FIGURE 5.4: The Deployed Network.

TABLE 5.2: Simulation Setup.

Parameter Name	Values
No of Nodes	50
Initial Energy	4000J
Deployment Area	1000x1000m
Simulation Tool	MATLAB R2021a
Routing Protocol	GRACE
Data Rate	50 Kbps

5.5 Results and Discussion

In this section, we discuss the results of our technique and then present the evaluation. In Figure 5.4, the initial deployment of 50 nodes is presented in 1000 mx1000 m area. Every node has a x co-ordinate and a y co-ordinate. The connected nodes are represented by a dotted line between those nodes. Whereas Table 5.3 shows the

TABLE 5.3: Packets allocated to different nodes

Number of Source Nodes	Number of Packets Generated Per Source
19	$Packets > 50$
14	$40 < Packets < 50$
12	$30 < Packets < 40$
5	$Packets < 30$

number of packets sent by different source nodes. The values depict a diversified packet generation at different nodes in order to imitate a realistic sensor network environment. As discussed in dataset section above, there were more than 4000 readings of node energy level from simulation environment. Figure 5.5 visualizes nodes and energy state information of the sensor nodes at network initialization. i.e start time reading. There are five different categories of node energy levels, shown with the help of five different colors. The green color represents the nodes with the maximum energy level, while the black color represents the nodes with the minimum energy level. These different color variation helps a network manager in

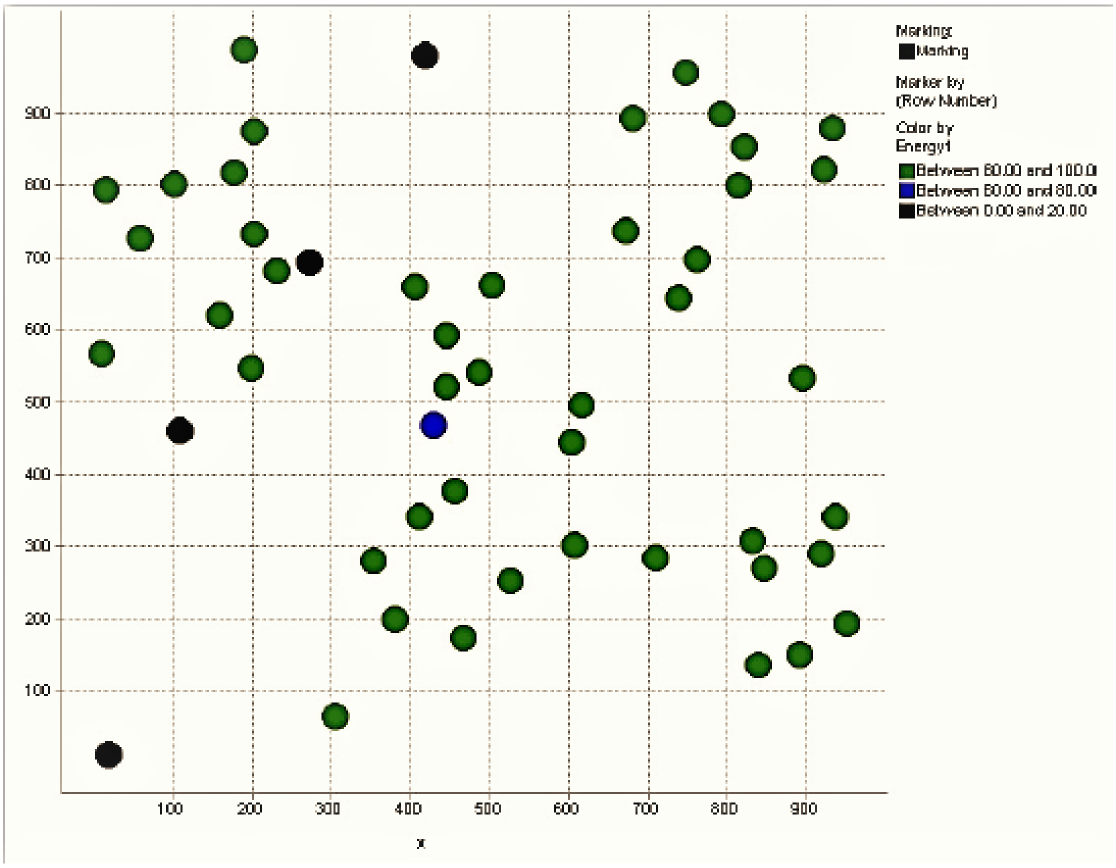

FIGURE 5.5: Initial Energy Visualization.

identifying vulnerable sensor nodes, which can cause potential cut or isolated regions in the network and assist him in taking re-deployment decisions to optimize network connectivity. Figure 5.6 depict the network scenario after certain time period of initial deployment. It can be observe, that some of the sensor nodes has exhausted their energy resources and can no longer actively participate in network operations and are thus, considered to be died (black nodes). whereas other sensor nodes energy reduces according to their usage (yellow,red,blue). This is the most critical phase, where a network manager can identify critical nodes which leads to partitioning and cuts in the network and take appropriate redeployment decisions to heal the network accordingly.

Figure 5.7 and Figure 5.8 show that our technique can successfully detect the isolated partition and network cuts problem using visualization. Thus, a WSN administrator can easily detect the problem and then take some countermeasures to ensure connectivity and avoid network partitioning. In Figure 5.11, we can

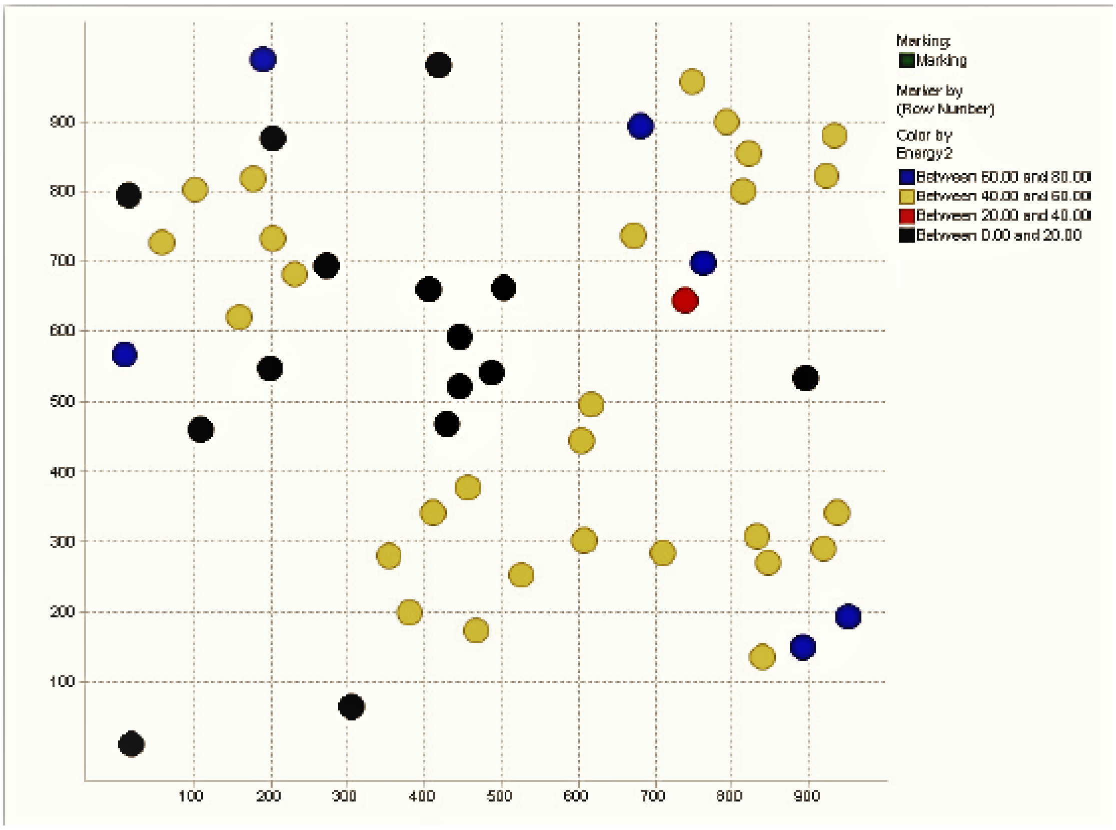

FIGURE 5.6: Energy Visualization after a certain time Period.

observe the state of the network when most sensors have exhausted their energy resources and the network is about to fail. Additionally, Figure 5.9 and Figure 5.10 illustrate the energy distribution of the entire network, depicted as bars. These energy values reflect the initial energy levels of the sensor nodes at the network initialization stage, as well as the energy status at the end of the simulation when a significant portion of the nodes have depleted their energy, signaling the network's imminent demise. the visual representation of energy distribution in the form of bar charts offers valuable insights into the energy consumption patterns within the network.

In a nutshell, By analyzing node energy visualization, the network manager can identify vulnerable nodes that have depleted their energy rapidly and are at a high risk of imminent failure. This knowledge is crucial as it allows the network manager to proactively address the situation, preventing network partitioning and potential disruptions in communication. By promptly replacing or recharging

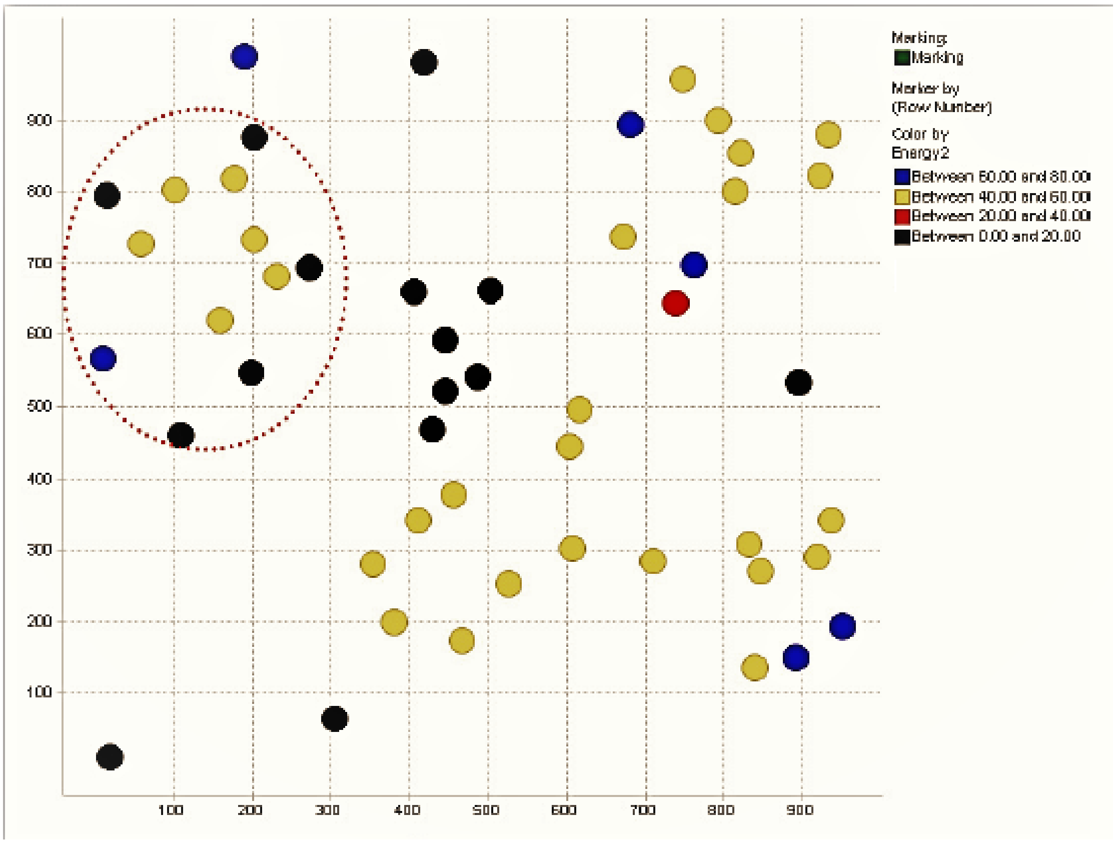

FIGURE 5.7: The Isolated Partition due to Dead Nodes.

these vulnerable nodes, the network's overall stability and connectivity can be maintained, ensuring seamless operation and minimizing the risk of data loss.

5.6 Evaluation

To evaluate our system, we conducted a detailed user study. We showed the outputs of our visualizations to domain experts with varying degrees of knowledge. We then examined the feedback from the experts. The assessment was based on two important factors: 1) identifying vulnerable nodes whose failure leads to partitioning of the network; and 2) is it necessary at a given time to redeploy additional sensor nodes in the network to ensure connectivity? The expert team is made up of people from different universities who are experts in sensor networks. The experts are divided into three categories and weighted differently according to their expertise, such as highly experienced, moderately experienced, and novice,

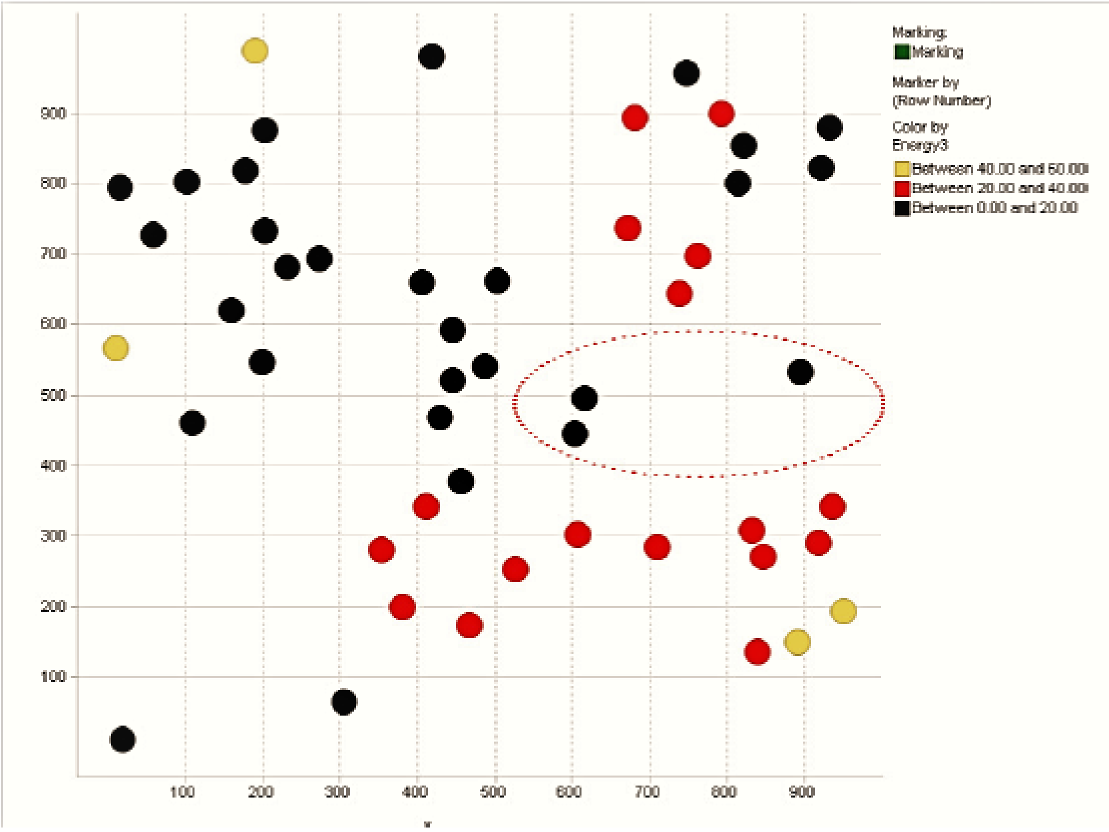

FIGURE 5.8: The Network Cut Created due to Dead Nodes.

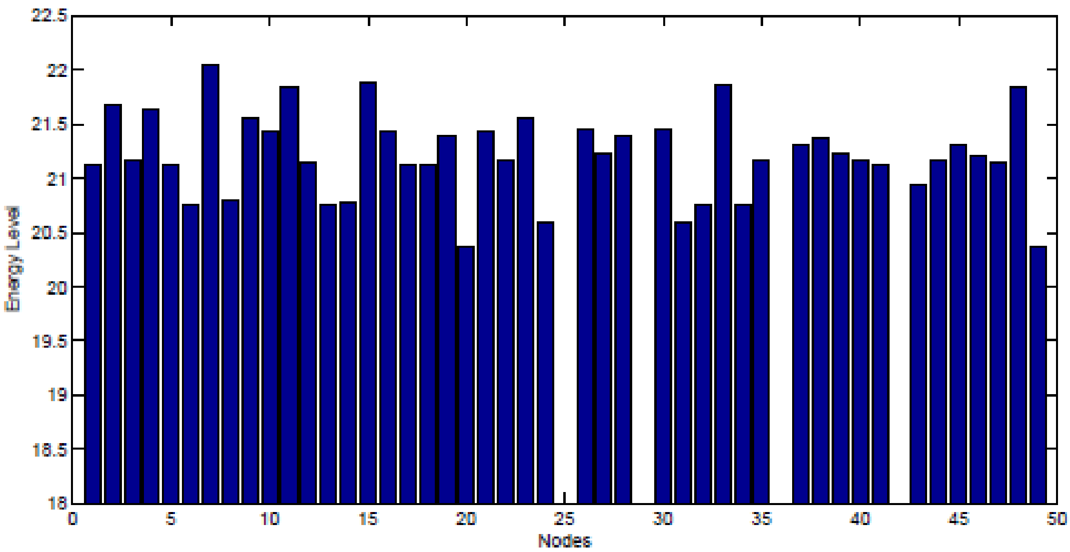

FIGURE 5.9: Energy levels of nodes at the Initialization.

as shown in Table 5.4. In the highly experienced expert category, the selected experts had either a Ph.D. in WSN or a Master's degree (MS) and also had extensive

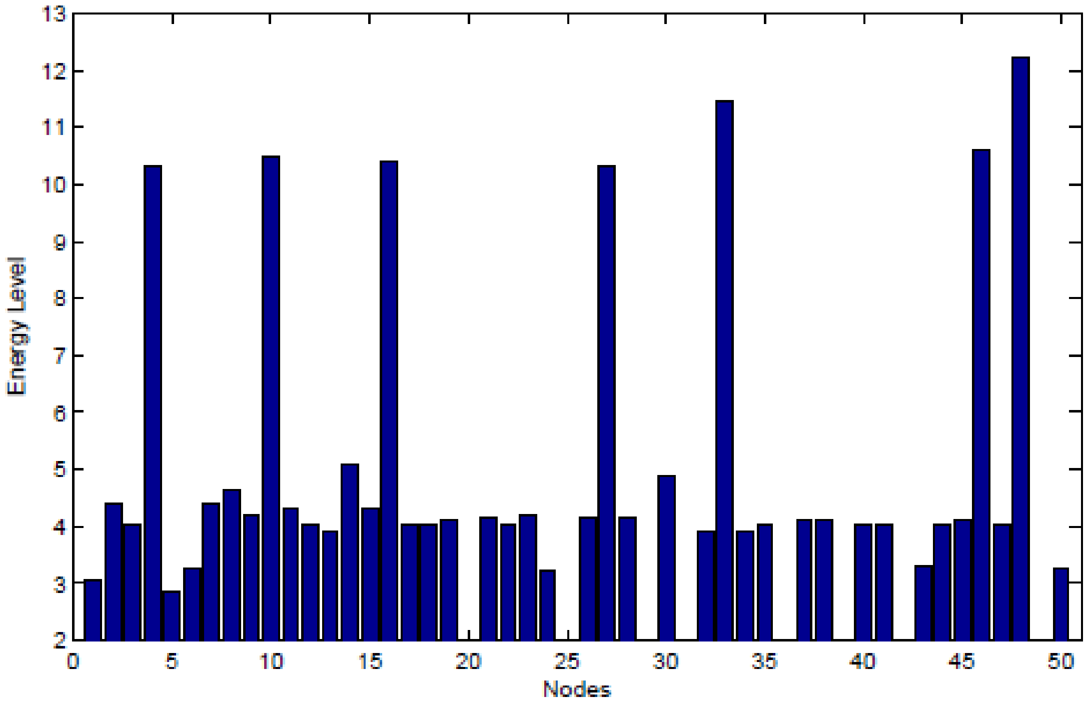

FIGURE 5.10: Energy levels of nodes at the End

TABLE 5.4: Experts Profile who Evaluated the Technique

S.No	Qualification	Overall Experience	Weight	Expertise Level
1	Ph.D	12	3	High
2	Ph.D	10	3	High
3	Ph.D	8	3	High
4	MS	10	3	High
5	MS	6	2	Medium
6	MS	5	2	Medium
7	MS	5	2	Medium
8	BS	1	1	Low
9	BS	1	1	Low
10	BS	1	1	Low

expertise in the field. The category of medium expert includes participants with a Master's degree and some experience in the field of WSN. The final category consists of users who has BS degree and enrolled in MS Program doing his research in wireless networks. Each evaluator is provided with network screenshots depicting

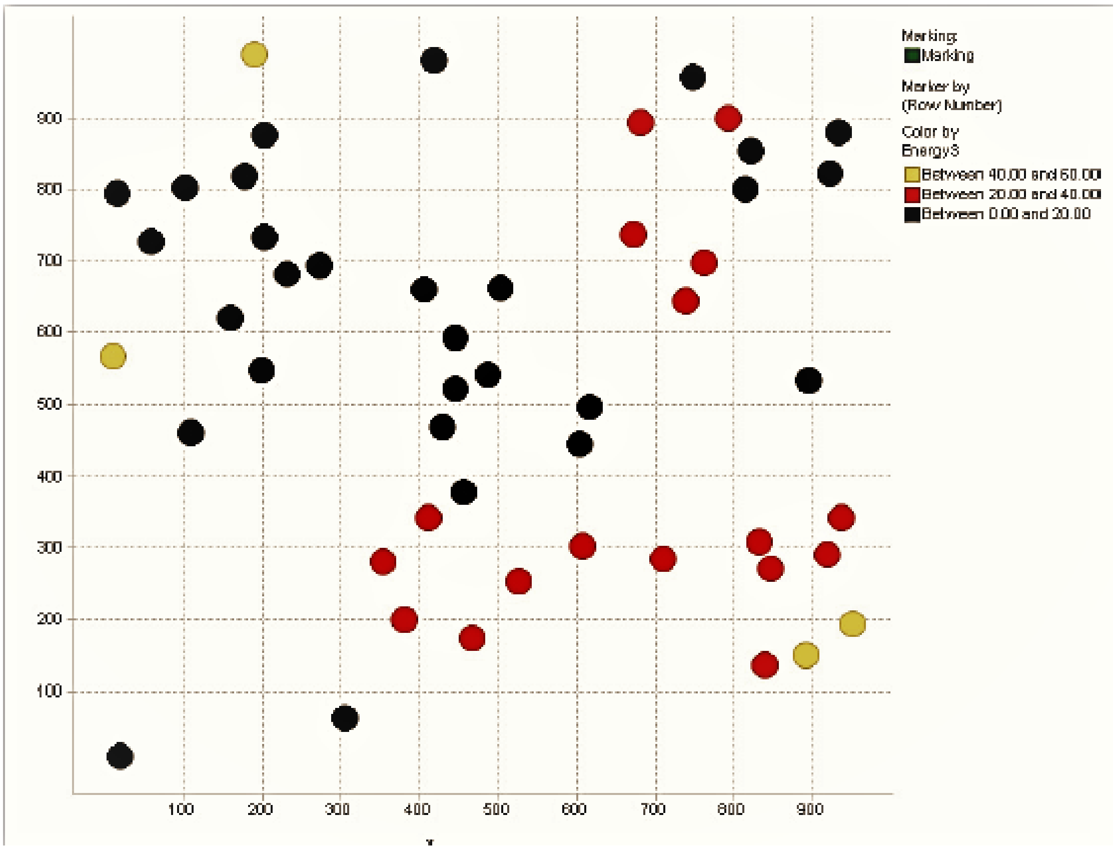

FIGURE 5.11: Energy Visualization at the End.

ten different scenarios. These network situations were chosen from several time occurrences, ranging from the best case to the worst. In addition, we capture screenshots for some of the scenarios that may result in network disconnection in the form of a cut or partition. After distributing such screenshots, we requested their judgement on identifying vulnerable nodes and whether or not we should re-deploy sensor nodes in the area of interest under the given circumstances to avoid network partitioning.

To incorporate the expertise level of each evaluator we have considered a weighted average response E_{node_i} of all the experts against each node in each scenario as Eq. 5.1

$$E_{node_i} = \begin{Bmatrix} 1 & if \quad \frac{\sum_{i=1}^{10} W_i E_i}{\sum_{i=1}^{10} W_i} > 0.60 \\ 0 & otherwise \end{Bmatrix} \tag{5.1}$$

where W_i denotes the weights assigned to the evaluator based on their level of expertise and E_i denotes the individual response from each evaluator. In a specific network scenario, feedback against each node was considered "1" (vulnerable node) if the weighted average response was greater than 60%, otherwise "0" (non-vulnerable node).

A confusion matrix Figure 5.12 is used to objectively analyze the feedback received to get a better understanding of the efficiency of the visualization, i.e., how helpful our visualization is in assisting the network manager to make accurate predictions about the vulnerability of the node and decide if it needs to be replaced or not. The confusion matrix consists of four basic features (numbers) used to define the

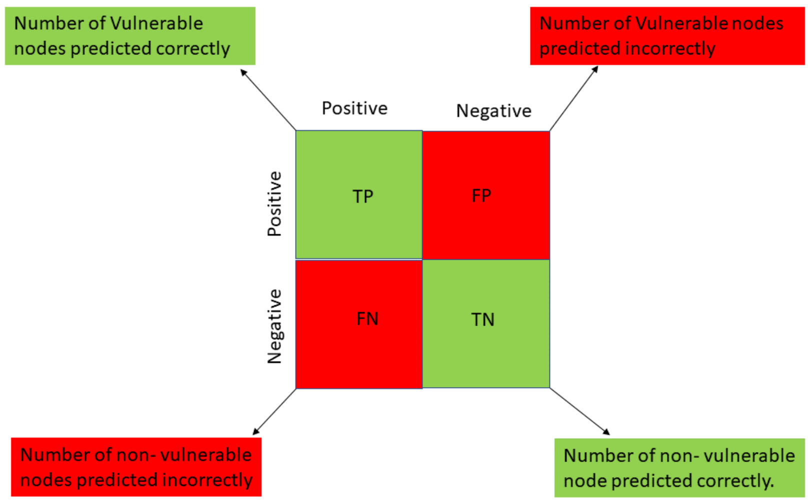

FIGURE 5.12: Confusion Matrix.

measurement metrics of the classifier (in our case, an expert) . These four numbers are:

1. TP(True Positive) : Represents the number of correctly predicted vulnerable nodes that need to be replaced to maintain connectivity

2. FP(False Positive) : Number of nodes that were incorrectly predicted to be vulnerable and they did not cause network fragmentation.

3. FN(False Negative) : Number of nodes that experts assumed did not need to be replaced, but whose failure caused network fragmentation.

4. TN(True Negative) : Represents the number of healthy/non-vulnerable nodes that were correctly predicted

Based on these values, the most commonly used performance metrics, Recall, Precision, Accuracy, and F1Score are calculated to evaluate the efficiency of the proposed visualization technique presented in Table 5.5.

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

$$F1Score = \frac{2 * precision * recall}{precision + recall} \tag{5.5}$$

Figure 5.13 shows the overall accuracy of the classifier, in our case the team of experts. We can see that the overall accuracy is mostly above 90%, indicating that our visualization is quite efficient in assisting the network managers to predict vulnerable nodes and recommend timely redeployment of additional sensor nodes to ensure connectivity. Figure 5.14, on the other hand, shows the F-score, a harmonic mean of precision and recall, which balances the result of uneven class distribution.

TABLE 5.5: Experts feedback Evaluation

Network Scenario	Recall	Precision	Accuracy	F1Score
1	0.83	0.88	0.90	0.85
2	0.76	0.90	0.92	0.83
3	0.88	0.93	0.94	0.90
4	0.94	0.88	0.94	0.91
5	0.88	0.88	0.96	0.88
6	0.80	0.92	0.92	0.85
7	0.93	0.93	0.96	0.93
8	0.77	0.87	0.88	0.82
9	0.88	0.93	0.93	0.90

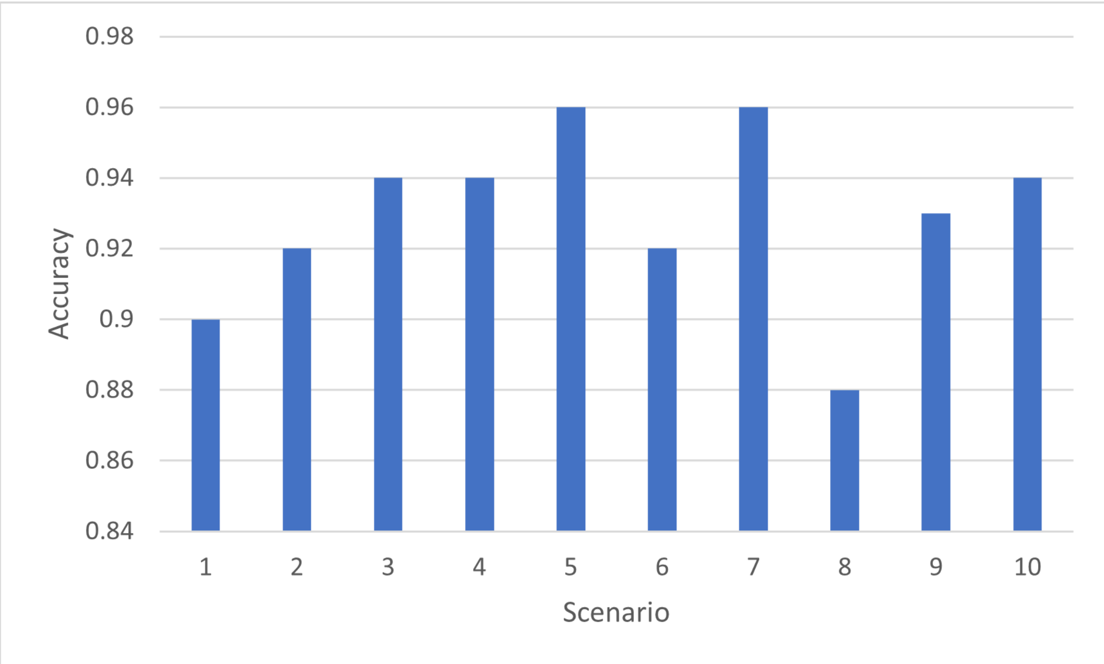

FIGURE 5.13: Accuracy of the identification of vulnerable nodes by Experts.

5.7 Conclusion

This chapter proposes a human assisted solution for detecting vulnerable sensor nodes and identifying potential partitions caused by node failure. The proposed solution relies on the visualization of the energy state information of sensor nodes to detect their vulnerability for their prospective failure. This information can be

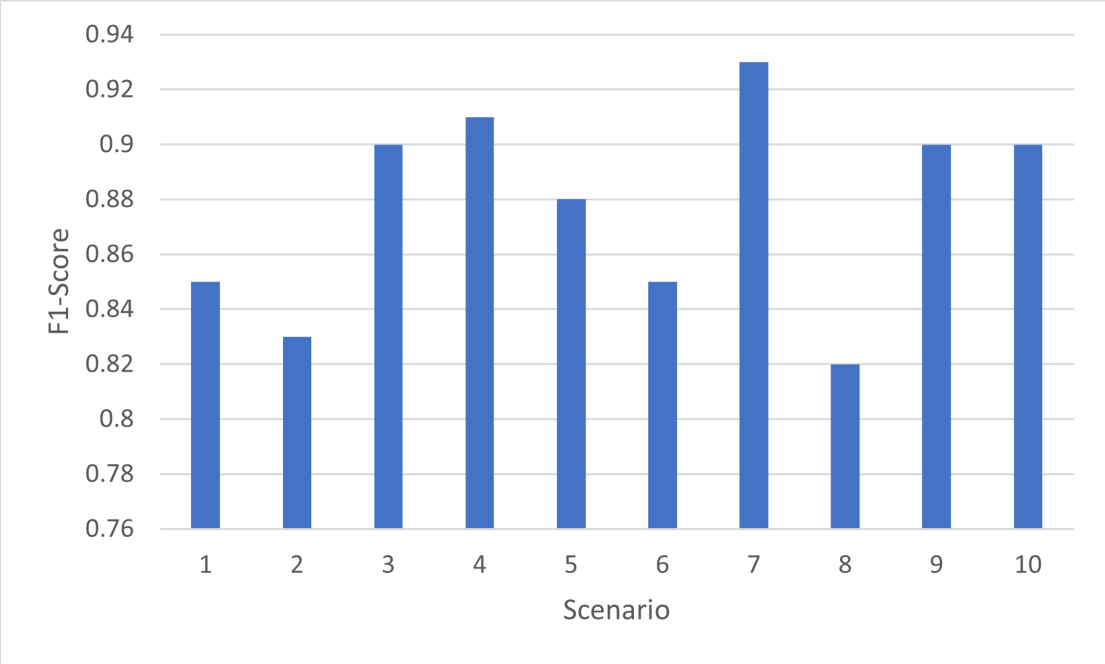

FIGURE 5.14: F1Score of the identification of vulnerable nodes by Experts.

used by network managers to predict and prevent possible network fragmentation. The proposed visualization technique also assists network managers in making redeployment decisions by identifying the exact location and instance to redeploy additional sensor nodes to ensure connectivity. The proposed visualization technique was evaluated by ten experts and their feedback was used to calculate three performance metrics, namely precision, recall, and accuracy, which shows that the proposed technique can adequately assist a network manager in partition prevention and redeployment decision-making.

Chapter 6

Conclusion and Future Work

6.1 Thesis Conclusion

The maintenance of connectivity and coverage is an important aspect of wireless sensor networks. It has a significant impact on the QoS aspects of the WSN. The primary task of any wireless sensor network is to monitor their area of interest for event occurrences and report them back to a central repository called a sink, which is only possible if a valid path exists. In this research work, we have investigated the connectivity issues of wireless sensor networks in detail and observed that the failure of a critical sensor node located along the path to the sink can lead to network partitioning. This may prevent a group of active nodes from effectively communicating time-critical data to the sink, which leads to network underutilization and causes the application to malfunctioning. We have proposed solutions for this problem in three different ways. The first proposed solution is through the traditional network topological information-based approach, where the sink node through its knowledge of network topology and sensor nodes' state information identifies the partition occurrences and proposes appropriate redeployment. The second solution is non-traditional and uses cooperative beamforming. while the third solution is based on energy visualization.

Our proposed traditional solution named as Partition Identification and Redeployment Algorithm (PIRA) starts with the initialization phase, where sensor node localization is performed by the sink node through setup messages from each sensor node in the network. Once the network is operational, all the sensor nodes keep track of their neighbors, such that if they detect any unresponsive sensor node in their neighborhood they report it to the sink and assume it to be dead. The sink then determines whether the failed node has caused a partition based on the information collected during the initial setup phase and received from the neighboring reporting nodes. Furthermore, the re-connectivity of these isolated partitions is achieved through the optimized deployment of the additional relay nodes taking into consideration the realistic 3D environment. The performance of the proposed technique was evaluated using different simulation parameters and compared with those of the notable existing solutions. The results show that the proposed technique, PIRA outperforms the existing strategies in terms of the number of detected partitions, duration of the partition detection, and the energy consumption of the sensor nodes during partition detection. The proposed technique not only detects network partitioning with low latency and energy consumption but also causes the lowest possible overhead in comparison with other techniques. As for as the limitations of the proposed technique PIRA are concerned, it has been observed that PIRA is not designed to handle partitions caused during the deployment phase or in scenarios where the network state information is not maintained. It is effective for detecting partitions in operational wireless sensor networks. It works well when the network is initially fully connected, and the sink node maintains network state information. Moreover, PIRA requires additional time for recovery through static node redeployment. To address the limitations of PIRA and to support real-time WSN applications, we have proposed a cooperative beamforming-based method as a second solution. The proposed Partition Identification and Reconnectivity through Cooperative Beamforming (PIRCB) aims to detect partitions and restore connectivity in near real-time. Thus, allowing us to retrieve valuable time-sensitive data from the nodes trapped in the disconnected partitions. In this method, we have considered both partitioning scenarios that occur during deployment and those that occur during network operation as a result of node failures. The proposed PIRCB allows sensor nodes trapped in isolated partitions to work together to create a directional beam that significantly extends their overall communication range. This allows them to communicate with a remote relay node connected to the core network. After undertaking extensive simulations, it has been observed that the proposed mechanism increases the fault-tolerance capability of a wireless sensor network without requiring additional resources. It also performs partition detection and re-connectivity in a timely manner. The simulation results show that under favorable conditions, the proposed PIRCB can heal up to 70% partitions in almost real time, thus providing a valuable solution for real-time WSN applications.

In order to devise a preemptive strategy to avoid network partitioning, we have proposed a third solution based on energy visualization. The proposed solution is designed to assist the network manager in identification of critical nodes which may lead to prospective partitions. The proposed technique relies on the visualization of sensor node energy state information, thus, enabling the network manager to proactively detect critical nodes and redeploy additional sensor nodes in their places to avoid network partitioning. In order to evaluate the effectiveness of our visualization technique, we have conducted a comprehensive user study involving field experts. We have sought their feedback in identifying vulnerable nodes that could potentially result in network partitioning. The feedback has been objectively analyzed using a confusion matrix to check the efficiency of the proposed visualization in assisting network managers. The study was focused on assessing the usefulness of the proposed visualization in accurately predicting node vulnerability and need for replacement. The results revealed an impressive overall accuracy rate of over 90 %. This indicates that the proposed visualization is highly effective in helping network managers to identify vulnerable nodes and make timely decisions regarding the deployment of additional sensor nodes to prevent network partitioning.

6.2 Future Work

As a future direction, the research conducted within this thesis possesses the potential for extension by incorporating the node vulnerability index in relation to energy depletion. This extension could be achieved through the development of a partition-aware routing protocol, which prioritizes the vulnerability of sensor nodes to conserve energy within the most critical nodes, specifically for the transmission of time-critical data. The realization of this objective would involve the computation of criticality index and connectivity impact costs for nodes during network operation. Such an approach would not only augment the longevity of the network, but will also curtail the formation of partitions.

The proposed cooperative beamforming method can be extended to address the localization of remote nodes situated within the network, which typically remain inaccessible through hop-by-hop communication within networks comprising general-purpose sensor nodes.

The effectiveness of cooperative beamforming relies upon the relative positioning and quantity of participating sensor nodes. Consequently, its potential can be further optimized through the development of a scheduling algorithm that optimally selects collaborating sensors based on considerations such as residual energy and relative position in relation to a reference node. Furthermore, the integration of network conditions and channel characteristics with the proposed algorithm can facilitate the dynamic adjustment of beamforming parameters, thereby enhancing performance. Moreover, network coding can be used to minimize the communication overhead on relay nodes.

The reconnectivity through cooperative beamforming can be integrated into traditional neighbor information based partition identification techniques to obtain a hybrid approach which can support real time wireless sensor applications.

In addition to energy visualization of sensor nodes, an expanded scope could encompass the provision of more advanced and intuitive representations of sensor data. This augmentation may include the utilization of three-dimensional visualization techniques, as well as interactive graphical depictions that effectively convey sensor analytics and data flow patterns in a more sophisticated manner.

The proposed energy visualization mechanism can be further refined through the incorporation of machine learning and artificial intelligence algorithms, enabling the development of sophisticated prediction models that automatically detect network anomalies and perform recovery without the need for human intervention, facilitating the identification of high-risk nodes and aiding the sink node in relocating mobile sensor nodes to prevent any potential network partitioning.

Furthermore, the proposed sensor network visualization can be utilized to detect security anomalies within the network, including the identification of compromised nodes that may be responsible for the creation of wormholes within the network.

Bibliography

[1] A. De La Piedra, F. Benitez-Capistros, F. Dominguez, and A. Touhafi, "Wireless sensor networks for environmental research: A survey on limitations and challenges," in *Eurocon 2013*. IEEE, 2013, pp. 267–274.

[2] R. M. Curry and J. C. Smith, "A survey of optimization algorithms for wireless sensor network lifetime maximization," *Computers & Industrial Engineering*, vol. 101, pp. 145–166, 2016.

[3] A. Ahir and H. Rakhunde, "An efficient approach towards recovery from node failure in wireless sensor network," *Int. J. Innov. Res. Sci. Technol*, vol. 2, pp. 3–5, 2015.

[4] I. F. Akyildiz and M. C. Vuran, *Wireless sensor networks*. John Wiley & Sons, 2010.

[5] R. C. Motta, V. Silva, and G. H. Travassos, "Towards a more in-depth understanding of the iot paradigm and its challenges," *Journal of Software Engineering Research and Development*, vol. 7, pp. 3–1, 2019.

[6] F. Pereira, R. Correia, and N. B. Carvalho, "Comparison of active and passive sensors for iot applications," in *2018 IEEE Wireless Power Transfer Conference (WPTC)*. IEEE, 2018, pp. 1–3.

[7] B. Akgül, M. Hasoğlu, and B. Haznedar, "Investigation and implementation ultra-low power pic based sensor node network with renewable energy source and decision-making unit," *Wireless Sensor Network*, vol. 10, 2018.

[8] M. Pathak, "An approach to memory management in wireless sensor networks," *Int J Comput Sci Eng Technol*, vol. 4, no. 08, pp. 1171–1176, 2013.

[9] P. K. Chong and D. Kim, "Surface-level path loss modeling for sensor networks in flat and irregular terrain," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, pp. 1–32, 2013.

[10] Y. Huang, J.-F. Martínez, J. Sendra, and L. López, "The influence of communication range on connectivity for resilient wireless sensor networks using a probabilistic approach," *International Journal of Distributed Sensor Networks*, vol. 9, no. 9, p. 482727, 2013.

[11] K. Zhu, "Ad hoc network deployment accommodating short and uncertain transmission range," in *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, 2011, pp. 101–108.

[12] T. K. Jain, D. S. Saini, and S. V. Bhooshan, "Lifetime optimization of a multiple sink wireless sensor network through energy balancing," *journal of Sensors*, vol. 2015, 2015.

[13] M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali, and H. Zain Eldin, "Deployment techniques in wireless sensor networks, coverage and connectivity: A survey," *IEEE Access*, vol. 7, pp. 28 940–28 954, 2019.

[14] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, 2005.

[15] R. E. Mohemed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks," *Computer Networks*, vol. 114, pp. 51–66, 2017.

[16] R. Baha and S. Ullash, "Detecting and healing area coverages holes in homogeneous wireless sensor network: Survey," *arXiv preprint arXiv:2005.02492*, 2020.

[17] R. Chowdhuri and M. K. D. Barma, "Node position estimation based on optimal clustering and detection of coverage hole in wireless sensor networks using hybrid deep reinforcement learning," *The Journal of Supercomputing*, pp. 1–33, 2023.

[18] K. M. Al-Aubidy, A. M. Derbas, and A. W. Al-Mutairi, "Real-time patient health monitoring and alarming using wireless-sensor-network," in *2016 13th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2016, pp. 416–423.

[19] D. Choudhary, R. Kumar, and N. Gupta, "Real-time health monitoring system on wireless sensor network," *International Journal of Advance Innovations, Thoughts & Ideas*, vol. 1, pp. 37–43, 2012.

[20] R. R. Kneipfer, "Sonar beamforming—an overview of its history and status," *Naval Undersea Warfare Center Detachment, Technical Report*, vol. 10, no. 003, pp. 1–20, 1992.

[21] C. A. Balanis, *Antenna theory: analysis and design.* John wiley & sons, 2015.

[22] S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "Distributed and collaborative beamforming in wireless sensor networks: Classifications, trends, and research directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2092–2116, 2017.

[23] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4110–4124, 2005.

[24] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004.* IEEE, 2004, pp. 81–88.

[25] R. Spence, *Information visualization: An introduction.* Springer, 2014.

[26] D. A. Keim, "Information visualization and visual data mining," *IEEE transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[27] E. Karapistoli and A. A. Economides, "Wireless sensor network security visualization," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*. IEEE, 2012, pp. 850–856.

[28] Surge, "Surge: Data visualisation tool. (accessed: 01.12.2021)." [Online]. Available: https://surge.my/our-services/data-stories-data-visualisation

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

[30] V. Ranga, M. Dave, and A. K. Verma, "Network partitioning recovery mechanisms in wsans: a survey," *Wireless personal communications*, vol. 72, no. 2, pp. 857–917, 2013.

[31] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[32] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," *IEEE transactions on mobile computing*, vol. 3, no. 3, pp. 272–285, 2004.

[33] J. Kleinberg, "Detecting a network failure," *Internet Mathematics*, vol. 1, no. 1, pp. 37–55, 2004.

[34] N. Shrivastava, S. Suri, and C. D. Tóth, "Detecting cuts in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, pp. 1–25, 2008.

[35] H. Ritter, R. Winter, and J. Schiller, "A partition detection system for mobile ad-hoc networks," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*. IEEE, 2004, pp. 489–497.

[36] P. Barooah, "Distributed cut detection in sensor networks," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 1097–1102.

[37] P. Barooah, H. Chenji, R. Stoleru, and T. Kalmár-Nagy, "Cut detection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 483–490, 2011.

[38] A. Zear and V. Ranga, "Distributed partition detection and recovery using uav in wireless sensor and actor networks," *Kuwait Journal of Science*, vol. 48, no. 4, 2021.

[39] S. M. Koriem and M. A. Bayoumi, "Detecting and measuring holes in wireless sensor network," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 8, pp. 909–916, 2020.

[40] D. E. Anna *et al.*, "Identifying partitions in wireless sensor network," *International Journal of Parallel Programming*, vol. 48, no. 2, pp. 296–309, 2020.

[41] V. Rao *et al.*, "Detection of connected but a cut occurred somewhere (ccos) in wireless sensor networks," *IJRCCT*, vol. 4, no. 6, pp. 410–413, 2015.

[42] K. Akkaya and M. Younis, "C2ap: Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," in *2007 IEEE International Performance, Computing, and Communications Conference*. IEEE, 2007, pp. 281–288.

[43] S. P. R. Banoth, P. K. Donta, and T. Amgoth, "Target-aware distributed coverage and connectivity algorithm for wireless sensor networks," *Wireless Networks*, pp. 1–16, 2023.

[44] V. Akbarzadeh, A. H.-R. Ko, C. Gagné, and M. Parizeau, "Topography-aware sensor deployment optimization with cma-es," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2010, pp. 141–150.

[45] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M. A. Mostafavi, "Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage," *IEEE transactions on instrumentation and measurement*, vol. 62, no. 2, pp. 293–303, 2012.

[46] M. Argany, M. A. Mostafavi, F. Karimipour, and C. Gagné, "A gis based wireless sensor network coverage estimation and optimization: A voronoi approach," in *Transactions on Computational Science XIV*. Springer, 2011, pp. 151–172.

[47] M. Argany, M. A. Mostafavi, V. Akbarzadeh, C. Gagné, and R. Yaagoubi, "Impact of the quality of spatial 3d city models on sensor networks placement optimization," *Geomatica*, vol. 66, no. 4, pp. 291–305, 2012.

[48] A. H. Ko and F. Gagnon, "Process of 3d wireless decentralized sensor deployment using parsing crossover scheme," *Applied Computing and Informatics*, vol. 11, no. 2, pp. 89–101, 2015.

[49] A. Zear, V. Ranga, and K. Bhushan, "Coordinated network partition detection and bi-connected inter-partition topology creation in damaged sensor networks using multiple uavs," *Computer Communications*, vol. 203, pp. 15–29, 2023.

[50] C.-F. Wang and J.-W. Ding, "The optimum sensor redeployment scheme using the most frangible clusters set," *Computer Communications*, vol. 31, no. 14, pp. 3492–3502, 2008.

[51] C. E. Otero, W. H. Shaw, I. Kostanic, and L. D. Otero, "Multiresponse optimization of stochastic wsn deployment using response surface methodology and desirability functions," *Systems Journal, IEEE*, vol. 4, no. 1, pp. 39–48, 2010.

[52] R.-S. Ko, "Analyzing the redeployment problem of mobile wireless sensor networks via geographic models," *Wireless Communications and Mobile Computing*, vol. 13, no. 2, pp. 111–129, 2013.

[53] R. Bhatt and R. Datta, "A stochastic process based framework of redeployment model for wireless sensor network," in *proc. Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, 2013, pp. 443–449.

[54] Z. Al Aghbari, P. P. Raj, and A. M. Khedr, "Ftcft: a fault-tolerant coverage preserving strategy for face topology-based wireless sensor networks," *The Journal of Supercomputing*, pp. 1–26, 2023.

[55] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, pp. 1–19, 2015.

[56] D. Wang, J. Jia, and L. Lin, "A cost-effective scheme for restoring connectivity of disconnected wsns," in *2015 International Conference on Information Networking (ICOIN)*. IEEE, 2015, pp. 69–74.

[57] R. Virender, D. Mayank, and V. A. Kumar, "Lost connectivity restoration in partitioned wireless sensor networks," in *Proceedings of International Conference on ICT for Sustainable Development*. Springer, 2016, pp. 89–98.

[58] Y. Zeng, L. Xu, and Z. Chen, "Fault-tolerant algorithms for connectivity restoration in wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 3, 2015.

[59] R. Kumar and T. Amgoth, "Adaptive cluster-based relay-node placement for disjoint wireless sensor networks," *Wireless Networks*, vol. 26, no. 1, pp. 651–666, 2020.

[60] J.-K. Zhang and D. Tao, "Partition connectivity recovery based on relay node deployment for wireless sensor networks," *Journal of Computers*, vol. 29, no. 6, pp. 193–200, 2018.

[61] Z.-G. Ding, D.-J. Du, and M.-R. Fei, "An isolation principle based distributed anomaly detection method in wireless sensor networks," *International Journal of Automation and Computing*, vol. 12, no. 4, pp. 402–412, 2015.

[62] M. Mansour and F. Jarray, "An iterative solution for the coverage and connectivity problem in wireless sensor network," *Procedia Computer Science*, vol. 63, pp. 494–498, 2015.

[63] A. A. Abbasi, K. Akkaya, and M. Younis, "A distributed connectivity restoration algorithm in wireless sensor and actor networks," in *32nd IEEE conference on local computer networks (LCN 2007)*. IEEE, 2007, pp. 496–503.

[64] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in wireless sensor network," in *2012 IEEE global communications conference (GLOBECOM)*. IEEE, 2012, pp. 652–657.

[65] M. Younis, S. Lee, S. Gupta, and K. Fisher, "A localized self-healing algorithm for networks of moveable sensor nodes," in *IEEE GLOBECOM 2008-2008 IEEE global telecommunications conference*. IEEE, 2008, pp. 1–5.

[66] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258–271, 2009.

[67] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*. IEEE, 2010, pp. 679–683.

[68] H. Essam, M. Younis, and E. Shaaban, "Minimum cost flow solution for tolerating multiple node failures in wireless sensor networks," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 6475–6480.

[69] Y. K. Joshi and M. Younis, "Exploiting skeletonization to restore connectivity in a wireless sensor network," *Computer communications*, vol. 75, pp. 97–107, 2016.

[70] G. Rajeswari, M. Sandhya, and K. Murugan, "Obstacle-aware connectivity restoration for the partitioned wireless sensor networks using mobile data carriers," *Wireless Networks*, vol. 29, no. 4, pp. 1703–1720, 2023.

[71] A. Wichmann, T. Korkmaz, and A. S. Tosun, "Robot control strategies for task allocation with connectivity constraints in wireless sensor and robot networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1429–1441, 2017.

[72] O. Dagdeviren, V. K. Akram, and B. Tavli, "Design and evaluation of algorithms for energy efficient and complete determination of critical nodes for wireless sensor network reliability," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 280–290, 2018.

[73] V. Jha, N. Prakash, and A. K. Mohapatra, "Energy efficient model for recovery from multiple nodes failure in wireless sensor networks," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1459–1479, 2019.

[74] S. Chakraborty, N. K. Goyal, S. Mahapatra, and S. Soh, "A monte-carlo markov chain approach for coverage-area reliability of mobile wireless sensor networks with multistate nodes," *Reliability Engineering & System Safety*, vol. 193, p. 106662, 2020.

[75] U. Baroudi, M. Aldarwbi, and M. Younis, "Energy-aware connectivity restoration mechanism for cyber-physical systems of networked sensors and robots," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3093–3104, 2020.

[76] V. Ranga, M. Dave, and A. K. Verma, "A hybrid timer based single node failure recovery approach for wsans," *Wireless personal communications*, vol. 77, no. 3, pp. 2155–2182, 2014.

[77] A. A. Awan, M. A. Khan, A. N. Malik, S. A. A. Shah, A. Shahzad, B. Nazir, I. A. Khan, W. Jadoon, N. Shahzad, and R. Nawaz Jadoon, "Quality of service-based node relocation technique for mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[78] S. Shriwastav and D. Ghose, "Round-table negotiation for fast restoration of connectivity in partitioned wireless sensor networks," *Ad Hoc Networks*, vol. 77, pp. 11–27, 2018.

[79] W. Fang, X. Song, X. Wu, J. Sun, and M. Hu, "Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks," *Information Fusion*, vol. 41, pp. 25–36, 2018.

[80] G. Dini, M. Pelagatti, and I. M. Savino, "An algorithm for reconnecting wireless sensor network partitions," in *proc Wireless Sensor Networks.* Springer, 2008, pp. 253–267.

[81] Z. Mi, Y. Yang, and J. Yang, "Restoring connectivity of mobile robotic sensor networks while avoiding obstacles," *Sensors Journal, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[82] E. A. Devi and J. Manickam, "Detecting and repairing network partition in wireless sensor networks," in *proc. Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on.* IEEE, 2014, pp. 1338–1343.

[83] M. Guizhen, Y. Yang, Q. Xuesong, G. Zhipeng, L. He, and X. Xiangyue, "Distributed connectivity restoration strategy for movable sensor networks," *Communications, China*, vol. 11, no. 13, pp. 156–163, 2014.

[84] Y. K. Joshi and M. Younis, "Restoring connectivity in a resource constrained wsn," *Journal of Network and Computer Applications*, vol. 66, pp. 151–165, 2016.

[85] G. Ma, Y. Yang, X. Qiu, and Z. Gao, "Obstacle aware connectivity restoration for disjoint wireless sensor networks using a mix of stationary and mobile nodes," *International Journal of Distributed Sensor Networks*, vol. 2016, 2016.

[86] I. Senturk, K. Akkaya, and S. Janansefat, “Towards realistic connectivity restoration in partitioned mobile sensor networks,” *International Journal of Communication Systems*, vol. 29, no. 2, pp. 230–250, 2016.

[87] B. Khalifa, A. M. Khedr, and Z. Al Aghbari, “A coverage maintenance algorithm for mobile wsns with adjustable sensing range,” *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1582–1591, 2019.

[88] K. Mahmood, M. A. Khan, A. M. Shah, S. Ali, M. K. Saeed *et al.*, “Intelligent on-demand connectivity restoration for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[89] C. Yang, S. Yuling, W. Zhongyi, and H. Lan, “Connectivity of wireless sensor networks in greenhouse for plant growth,” *International Journal of Agricultural and Biological Engineering*, vol. 9, no. 1, pp. 89–98, 2016.

[90] P. Kar, A. Roy, S. Misra, and M. S. Obaidat, “Energy-efficient connectivity re-establishment in wsn in the presence of dumb nodes,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 1485–1490.

[91] J. E. Agudo, J. F. Valenzuela-Valdés, F. Luna, R. M. Luque-Baena, and P. Padilla, “Analysis of beamforming for improving the energy efficiency in wireless sensor networks with metaheuristics,” *Progress in Artificial Intelligence*, pp. 1–8, 2016.

[92] M. F. Ahmed and S. Vorobyov, “Performance characteristics of collaborative beamforming for wireless sensor networks with gaussian distributed sensor nodes,” in *proc. Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3249–3252.

[93] M. F. Ahmed and S. A. Vorobyov, “Collaborative beamforming for wireless sensor networks with gaussian distributed sensor nodes,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 638–643, 2009.

[94] M. A. Fernandes, "Collaborative reception technique in wireless sensor networks using adaptive antennas," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–17, 2014.

[95] A. Ghani, H. Naqvi, M. Sher, A. U. Rahman, and I. Khan, "Capacity gain in spread spectrum based collaborative communication in wireless sensor networks," *Journal of Internet Technology*, vol. 20, no. 3, pp. 731–740, 2019.

[96] E. Jones, "Distributed cooperative sensor networks using intelligent adaptive antennas," in *proc. Geoscience and Remote Sensing Symposium, . IGARSS 2007. IEEE International.* IEEE, 2007, pp. 2931–2934.

[97] A. Kavak and K. Kucuk, "On connectivity analysis of smart antenna capable wireless sensor networks," in *proc. Wireless Communication Systems, 2009. ISWCS 2009. 6th International Symposium on.* IEEE, 2009, pp. 6–10.

[98] I. Thibault, G. E. Corazza, and L. Deambrogio, "Phase synchronization algorithms for distributed beamforming with time varying channels in wireless sensor networks," in *proc. Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International.* IEEE, 2011, pp. 77–82.

[99] B. Bejar Haro, S. Zazo, and D. P. Palomar, "Energy efficient collaborative beamforming in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 62, no. 2, pp. 496–510, 2014.

[100] B. Béjar, S. Zazo, and D. P. Palomar, "Lifetime maximization for beamforming applications in wireless sensor networks," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE, 2012, pp. 2849–2852.

[101] J. Feng, Y.-H. Lu, B. Jung, D. Peroulis, and Y. C. Hu, "Energy-efficient data dissemination using beamforming in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 3, p. 31, 2013.

[102] M. Gasmi, O. Mosbahi, M. Khalgui, L. Gomes, and Z. Li, "Performance optimization of reconfigurable real-time wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[103] N. Hassan, N. M. Khan, G. Ahmed, and R. Ramer, "Real-time gradient cost establishment (rt-grace) for an energy-aware routing in wireless sensor networks," in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing.* IEEE, 2013, pp. 54–59.

[104] C. H. Wong, Z. W. Siew, H. P. Yoong, A. Kiring, and K. T. K. Teo, "Node collaboration for distributed beamforming in wireless sensor networks," in *proc. Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on.* IEEE, 2012, pp. 483–488.

[105] D. To, T. To, and J. Choi, "Energy efficient distributed beamforming with sensor selection in wireless sensor networks," in *Proc. Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th.* IEEE, 2012, pp. 1–5.

[106] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, 2005.

[107] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," *The Journal of Supercomputing*, vol. 76, pp. 7333–7373, 2020.

[108] Z. A. Dagdeviren and V. Akram, "Connectivity estimation approaches for internet of things-enabled wireless sensor networks," in *Emerging Trends in IoT and Integration with Data Science, Cloud Computing, and Big Data Analytics.* IGI Global, 2022, pp. 104–122.

[109] H. Wang, X. Ding, C. Huang, and X. Wu, "Adaptive connectivity restoration from node failure (s) in wireless sensor networks," *Sensors*, vol. 16, no. 10, p. 1487, 2016.

[110] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 551–591, 2012.

[111] S. M. Chowdhury and A. Hossain, "Different energy saving schemes in wireless sensor networks: A survey," *Wireless Personal Communications*, vol. 114, no. 3, pp. 2043–2062, 2020.

[112] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[113] L. Dong, A. P. Petropulu, and H. V. Poor, "Cooperative beamforming for wireless ad hoc networks," in *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*. IEEE, 2007, pp. 2957–2961.

[114] L. Shi, Z. Li, X. Ding, J. Xu, and Z. Lv, "Optimizing wireless sensor networks based on collaborative beamforming," *Procedia Computer Science*, vol. 174, pp. 561–571, 2020.

[115] E. Chi, "A taxonomy of visualization techniques using the data state reference model," in *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, 2000, pp. 69–75.

[116] S. Ravichandran, R. K. Chandrasekar, A. S. Uluagac, and R. Beyah, "A simple visualization and programming framework for wireless sensor networks: Proviz," *Ad Hoc Networks*, vol. 53, pp. 1–16, 2016.

[117] C. Buschmann, D. Pfisterer, S. Fischer, S. P. Fekete, and A. Kröller, "Spyglass: a wireless sensor network visualizer," *Acm Sigbed Review*, vol. 2, no. 1, pp. 1–6, 2005.

[118] L. Shu, M. Hauswirth, H.-C. Chao, M. Chen, and Y. Zhang, "Nettopo: A framework of simulation and visualization for wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 5, pp. 799–820, 2011.

[119] E. Kelechi, I. Linus, and I. Obioma, "Network visualization: Application to wireless network," *Network*, vol. 4, no. 1, 2015.

[120] N. M. Khan, Z. Khalid, and G. Ahmed, "Gradient cost establishment (grace) for an energy-aware routing in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1–15, 2009.

[121] T. S. official. Tibco software inc. (accessed: 01.11.2021). [Online]. Available: https://www.tibco.com/products/tibco-spotfire.html