



Faculty of Sciences

Department of Computer Science and Information Systems

**OPTIMIZING DATA DELIVERY IN WIRELESS SENSOR
NETWORKS: USING ENHANCED SHORTEST-PATH
ALGORITHMS (MODIFIED DIJKSTRA & SPIN-G)**

by

Meshach Morlu

Student ID number: 19100145

DIT (Amity University)

BSc Honours (Network Security and Computer Forensics) (Botho University)

A Dissertation/Thesis Submitted to the Faculty of Sciences in Partial Fulfilment of the
Requirements for the Award of the Degree of Master of Science in Computer Science and
Software Engineering of BIUST

Supervisor: Prof. Rajalakshmi Selvaraj

Co-Supervisor: Prof. Venumadhav Kuthadi

Department of CSIS

Faculty of Science, BIUST

selvarajr@biust.ac.bw

kuthadiv@biust.ac.bw

November 30, 2023

DECLARATION REGARDING THE WORK AND COPYRIGHT

Candidate (please write in caps or type)

MESHACH MORLU

Student ID:

19100145

Thesis Titled: **RELIABLE SHORTEST PATH ALGORITHM FOR DATA DELIVERY TO IMPROVE THE QUALITY OF SERVICE IN WIRELESS SENSOR NETWORKS**

I, the **Candidate**, certify that the Thesis is all my own original work and that I have not obtained a degree in this University or elsewhere on the basis of any of this work.

(If the thesis is based on a group project, then the student must indicate the extent of her / his contribution, with reference to any other theses submitted or published by each collaborator in the project, and a declaration to this effect must be included in the thesis)

This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright and Neighbouring Rights Act, Act. No. 8 of 2000 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Postgraduate School, on behalf of both the author and the BIUST.

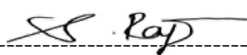


Signed: ----- Date: -----30 Nov 2023-----

Primary Supervisor (please write in caps or type)

Prof **RAJALAKSHMI SELVARAJ**

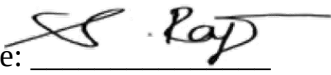
I, the Candidate's **Primary Supervisor**, hereby confirm that I have inspected the above titled thesis and, to the best of my knowledge, it is based on the original work of the candidate.

Signed:  ----- Date: -----30 Nov 2023-----

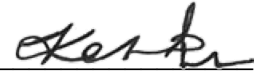
CERTIFICATION

This document signatory endorses that he or she has reviewed the document and firmly recommends for approval by the Faculty of Sciences a thesis titled: **Reliable Shortest Path Algorithm for Data Delivery to Improve the Quality of Service in Wireless Sensor Networks**, in partial fulfillment of the degree of Master of Science in Computer Science and Software Engineering of the Botswana International University of Science and Technology (BIUST).

Supervisor: Prof. Rajalakshmi Selvaraj

Signature: 

Co-Supervisor: Prof. Venumadhav Kuthadi

Signature: 

ACKNOWLEDGEMENT

I thank the Lord Almighty for the profound knowledge that enabled me to develop these research ideas that resulted in the successful completion of this thesis. Furthermore, I would like to express my heartfelt appreciation to my thesis supervisors, Prof. Selvaraj Rajalakshmi and Prof. Venumedhav Kithadi, for their advice, contributions, and knowledge that aided me in my academic career. Many thanks to Professor Adamu Murtala Zungeru and Dr. Thabo Semong for their intellectual insights into my research topic, as well as their constructive criticism, which helped me refine my thesis. My family, I just want to express my gratitude for your financial and moral support. Finally, I'd like to express my heartfelt appreciation to the Botswana International University of Science and Technology.

Abstract

Wireless sensor networks are made from the connection of networks of devices that cooperate using radio frequency for the sole purpose of communicating information acquired by the sensor node while monitoring and recording a certain condition of an event that might occur. Wireless sensor networks (WSNs) have grown rapidly in recent years, necessitating the need to improve the Quality of Service (QoS) in the Sensor Protocol for Information via Negotiation (SPIN), which this research has addressed. This research chose to concentrate on improving the data delivery rate in SPIN which will improve the QoS in WSNs. QoS is the service requirements that a particular technology should meet to operate under a certain desirable condition. Inventing wireless technologies gave birth to the SPIN protocol, which is a data-centric negotiation-based protocol for the smooth transmission of data among wireless devices. By combining the location-based mechanism in geography adaptive fidelity (GAF) with SPIN, this research transforms the broadcast mechanism in the SPIN protocol into a data-delivery mechanism. Via MATLAB simulator, the combination of the SPIN protocol with the GAF location-based algorithm creates a new protocol called SPIN-G, or Sensor Protocol for Information via Negotiation Geography, that enhances data delivery in wireless sensor networks. Moreover, integrating the GAF location-based algorithm into the SPIN protocol and incorporating a modified Dijkstra shortest path algorithm for data delivery in SPIN-G is an interesting approach. The goal of the modified Dijkstra algorithm implementation on SPIN-G is to maximize throughput while minimizing latency. To guarantee successful data delivery, the SPIN-G employs the energy-aware position-constructed routing technique. The merger of SPIN and GAF has led to the formation of a cluster-based topology for SPIN-G. To evaluate the protocols in this research, performance metrics like packet loss, packet delivery ratio, signal noise ratio, and end-to-end delay are used to calculate and rate the performance of SPIN, GAF, SPIN-G, and the modified Dijkstra algorithm in terms of the QoS for WSNs. By recording and analyzing these performance metrics, this research provides evidence of the improvements in QoS for SPIN-G compared to the original SPIN protocol. The reported 99.1% successful packet delivery rate for SPIN-G suggests a substantial enhancement in packet delivery efficiency, which indicates improved QoS for WSNs. Future research in WSNs should prioritize testing the SPIN-G protocol with different protocols and network topologies, assessing network lifetime and power consumption, and creating strategies to mitigate transmission failures, aiming to enhance the overall performance, efficiency, and reliability of WSNs for diverse applications.

Table of Contents

ACKNOWLEDGEMENT	iv
Abstract	v
Table of Contents.....	vi
List of Figures	viii
List of Tables	ix
List of Abbreviations	x
Chapter 1	2
1.0 Introduction.....	2
1.0.1 Wireless Sensor Networks (WSNs)	3
1.0.1.1 Sensor Node	4
1.0.2 Essence of Quality of Service in WSNs.....	5
1.0.3 Sensor Protocol for Information via Negotiation (SPIN).....	6
1.0.4 Geographic Adaptive Fidelity	7
1.0.5 Shortest Path Algorithm in WSNs	8
1.1 Problem Background and Related Research	9
1.2 Problem Statement	10
1.3 General Objective	10
1.3.1 Specific Objectives	10
1.3 Research Question	11
1.5 Contribution to Body of Knowledge.....	11
1.6 Research Motivation	12
1.7 Thesis Organization	12
Chapter 2	14
2.0 Literature Review.....	14
2.1 Wireless Sensor Networks	14
2.2 Wireless Sensor Network Design Approach.....	15
2.3 WSN Deployment Technique	17
2.4 Significance of Cluster-Based Topology in WSNs.....	18
2.5 Importance of Clustering in SPIN.....	19
2.6 Clustered Topology Implemented for SPIN	19
2.7 Clustered Topology Implemented for SPIN-G	20
2.8 Non-Clustered Head Architecture in SPIN-G.....	20
2.9 Data Delivery Model for WSN	21
2.10 Data Transmission Issues in SPIN	22
2.11 Data Delivery in SPIN-G	23
2.12 Essence of Geography Adaptive Fidelity in WSN.....	24

2.13 Shortest Path Algorithm and Drawbacks	25
2.14 Dijkstra Algorithm	26
2.15 Modified Dijkstra Algorithm	27
2.16 Quality of Service Necessities in Wireless Sensor Networks	29
2.17 Quality of Service Challenges in Wireless Sensor Networks	29
2.18 Quality of Service in SPIN.....	31
2.19 Quality of Service of SPIN-G	32
2.20 Performance Metrics for WSNs	32
2.21 Throughput in WSN.....	33
2.22 Throughput in SPIN-G.....	34
2.23 Overall Summary of Literature	35
Chapter 3	36
3.0 Methodology.....	36
3.0.1 To develop a protocol that maximizes data delivery by integrating SPIN and GAF location-based algorithms.	36
3.0.2 To implement cluster-based topology for the proposed SPIN-G.....	38
3.0.3 To implement the modified Dijkstra shortest path algorithm in SPIN-G	39
3.0.4 To construct performance metrics for SPIN, GAF, and SPIN-G protocol to check the QoS	41
3.0.5 To compare and evaluate the performance of the SPIN-G against the SPIN and GAF protocols.....	42
3.0.6 Implementation methods.....	43
3.0.0.1 Phase I: Sensor Nodes Transmission	44
3.0.0.2 Phase II: Data Delivery Mechanism using SPA	46
3.0.0.3 Phase III: Implementing the parameters	47
3.1 Flowchart for the implemented approach	47
3.2 Desirable approach.....	48
3.3 Performance Metrics	50
3.4 Overall Summary of Methodology	56
Chapter 4	58
4.0 Simulation Results and Discussion	58
4.1 Experimental Results	58
4.2 Performance Evaluation of Protocols.....	64
4.3 Packet Loss	67
4.4 Latency.....	69
4.5 Throughput.....	71
4.6 Conclusion	73
Chapter 5	74
5.0 Conclusion and Future Work	74
5.1 Future Improvements	75
References.....	77

List of Figures

Figure 1: WSNs infrastructure design.....	3
Figure 2: The architecture of a typical sensor node	4
Figure 3: Sensor device used in WSNs Infrastructure for acquiring data and transmitting to sink.	5
Figure 4: The QoS Model	6
Figure 5: Data transmission patterns in SPIN	7
Figure 6: State transition in GAF protocol.....	8
Figure 7: A graphical hierarchical outline of this thesis.	13
Figure 8: The classification of existing approaches for WSN design.	16
Figure 9: The flowchart diagram shows the four data-driven model states in WSN.	21
Figure 10: Three operation principles in SPIN using the broadcast mechanism	22
Figure 11: The implementation of Clustered-based topology for SPIN with Dijkstra Algorithm.....	44
Figure 12: This diagram shows the implementation of modified Dijkstra in SPIN-GAF with 50 nodes.	46
Figure 13: The flowchart diagram of the implementation of the proposed SPIN-G protocol.	47
Figure 14: Modified Dijkstra implemented in SPIN-G showing the interlink of sensors and packets sent from source to destination (Iteration 1).	59
Figure 15: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 2).	60
Figure 16: Modified Dijkstra implemented on SPIN-G showing the dead node and the hops (Iteration 3)	61
Figure 17: Modified Dijkstra implemented in SPIN-G showing the dead node and the hop count (Iteration 4)	62
Figure 18: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 5)	62
Figure 19: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 6)	63
Figure 20: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 7)	63
Figure 21: Modified Dijkstra implemented to SPIN-G capture no connection between source to the destination due to dead nodes and out-of-range nodes (Iteration 8).	64
Figure 22: Evaluating the QoS in WSNs through packet loss vs the number of sensor nodes.....	67
Figure 23: Using packet loss vs. packet delivery ratio to assess QoS in WSNs.	68
Figure 24: Evaluating the QoS in WSN through end-to-end delay vs the number of sensor nodes.	69
Figure 25: Evaluating the QoS in WSN through end-to-end delay vs the number of destinations.....	70
Figure 26: Using packet sent vs. Signal to Noise ratio to assess QoS in WSNs.....	71
Figure 27: The performance evaluation of signal noise ratio against the bit error rate.	72

List of Tables

Table 1: Comparison of QoS-aware routing protocols in WSN (Patel & Raja, 2014).	31
Table 2: Outline in this table are the initialized and output values for SPIN-G with Modified Dijkstra.	58
Table 3: Outline in this table are the initialized and output values for Modified Dijkstra with SPIN-G	60
Table 4: Outline in this table are the initialized and output values for Modified Dijkstra in SPIN-G..	61
Table 5: Performance check for GAF, SPIN, and SPIN-G protocols by evaluating packet loss against the number of sensor nodes.....	67
Table 6: Performance check for GAF, SPIN, and SPIN-G protocols by evaluating packet loss against packet delivery rate.	68
Table 7: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the end-to-end delay against the number of sensor nodes.....	69
Table 8: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the end-to-end delay against the number of destinations.	70
Table 9: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the packets sent to the base station against the number of destinations.	71
Table 10: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the packets sent to the base station against the number of destinations.	72
Table 11: This table shows the overall performance of three key parameters used in checking QoS.	73

List of Abbreviations

ADC: Analog to Digital Convertor

ADV: Advertisement message

BS: Base Station

CH: Cluster Head

EEHMCS: Energy-Efficient Hierarchical Multicast Communication System

FPGAs: Field Programmable Gate Arrays

GAF: Geography Adaptive Fidelity

P2P: Point-to-Point

QoS: Quality of Service

REQ: Data Request

SHMCS: SPIN-based Hybrid Multi-Hop Clustering Scheme

SPA: Shortest Path Algorithm

SPIN: Sensor Protocol for Information Via Negotiation

WSN: Wireless Sensor Networks

Chapter 1

1.0 Introduction

In this ubiquitous era of emerging technologies, wireless sensor networks (WSNs) are one of the innovative fields where there's competition between researchers and tech inventors. WSNs are a collection of circulated independent network components using sensors to supportively observe ecological or physical conditions, such as temperature, sound, vibration, pressure, motion, and pollutants, at different sites (Tiwari et al., 2015). The topology for WSNs contains several sensor nodes that are scattered in harsh environments (Walia & Mehta, 2015). It is necessary to enhance the sensor protocol for information via negotiation (SPIN) protocol because the quality of service (QoS) will be optimized since WSNs are seen as the wireless networks of the future.

To enhance the QoS in WSNs, this research will present novel concepts and solutions for the issue of data delivery in the SPIN protocol. The evaluation of the QoS in the SPIN protocol has given light on how this research can improve in key areas. Data delivery in the SPIN protocol is a key point of focus this research evaluated in wireless sensor networks. This research chooses to contribute by focusing on adding new components to ensure data delivery in the SPIN protocol for wireless sensor networks that enhance its functionalities. When SPIN was first developed, it used the broadcast technique to distribute packets among WSNs according to (Tiwari et al., 2015), (Walia & Mehta, 2015).

The SPIN is an indispensable protocol considering the key roles and contributions of WSNs. SPIN forms part of the data-centric group and it's an event-driven protocol. It is of the essence to conduct research and experiment on the SPIN routing protocol in WSNs because it is a protocol in WSNs whose main function is to conduct an efficient flow of operation, security, and energy conservation, and SPIN is available in almost every WSN architecture. The SPIN protocol is implemented in the WSNs architecture with the broadcast mechanism to disseminate the data. It falls in the data-centric negotiation-based category for WSNs. The design of the SPIN protocol contains a flaw that does not guarantee the trustworthy delivery of data; instead, data delivery is dependent on an event (Ali & Kumar, 2008).

The SPIN protocol in WSNs is a data-centric negotiation-based protocol noted for its broadcast system of data transmission. The main goal of the SPIN protocol is to effectively disseminate data gathered by independent sensor nodes to every other node in the network. Despite the transition from wired sensor networks to wireless sensor networks, there are still some challenges that do exist particularly in the protocols. However, the most obvious obstacles in the protocol could be quality of service, security, or interference that generates a bit error rate in transmission, as well as node mobility, which can result in frequent communication breakdowns (Oudani et al., 2016).

1.0.1 Wireless Sensor Networks (WSNs)

Wireless sensor networks (WSNs) are made up of several sensor nodes that can detect physical events, process the detected data locally, and then send the raw or aggregated data to a distant base station. In Figure 1 below, you can clearly view the parties involved in the complete WSNs setup. There are the sensor nodes, based station (BS), internet, and the user. The fundamentals of a Wireless Sensor Network's generic transport layer described the essential part of data transmission in the architecture. The Internet's transport layer manages the congestion brought on by network traffic as well as the end-to-end dependability of individual packets. Many WSN applications, like those on the Internet, require a congestion management mechanism to manage the volume of traffic pumped into the WSN to prevent packet loss and ensure end-to-end dependable packet delivery (Rahman et al. 2008). Figure 1 depicts the physical structure of WSNs and how devices connect between sensor nodes and the BS, which connects to the user via the internet.

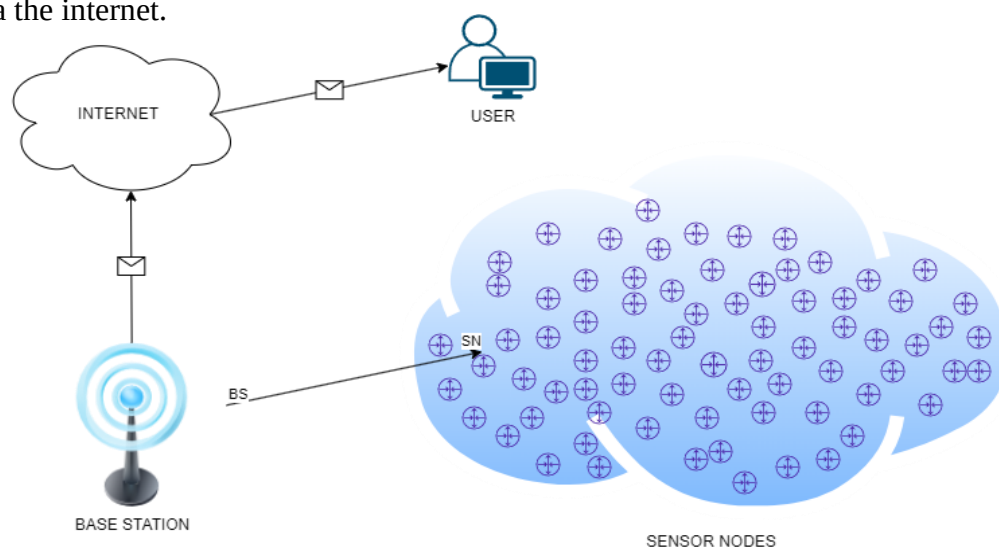


Figure 1: WSNs infrastructure design(Rahman et al. 2008).

1.0.1.1 Sensor Node

A sensor node is a device that perceives and gathers data for application-specific analysis for sensing, processing, and communication capabilities when it is employed in a network. There are many design considerations for a sensor network application, including sensing modality, sensor node computation, communication, and storage capabilities. Furthermore, the cost and size of each node, type of power source, architecture for deployment, protocols for data dissemination, communication, applications, and management tools, to mention a few (Sudevalayam & Kulkarni, 2011). Figure 2 depicts the various components of the sensor node architecture, including the analog/digital converter (ADC) (Kochhar et. al, 2018). In this architectural diagram, the sensor node is made up of a sensor, processor, communication module, memory, and power source. Every component serves an important purpose that necessitates collaboration in order to function properly.

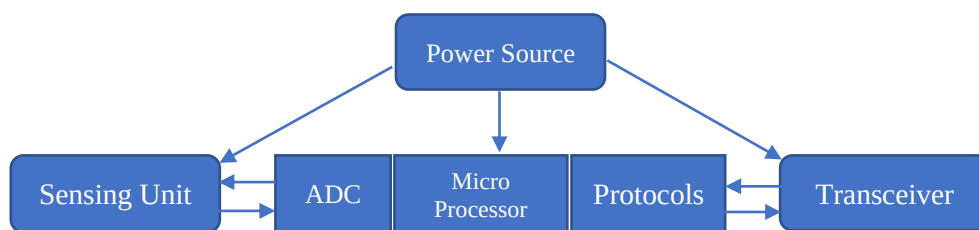


Figure 2: The architecture of a typical sensor node (Sudevalayam & Kulkarni, 2011).

Designing sensor networks is a complex task that requires careful consideration of a wide range of factors. The placement of sensor nodes, for example, can have a significant impact on the performance of the network, as it affects the connectivity and coverage of the sensors. In addition, factors like energy consumption and network lifetime are critical considerations, as sensor nodes are often powered by batteries and need to operate for extended periods without maintenance. Other factors, such as fault tolerance, quality of service, node mobility, and security, can also impact the performance and reliability of the sensor network. For example, fault tolerance is critical for ensuring that the network can continue to function even if some sensors fail or become unavailable (Adamu, 2012).

Quality of service is important for ensuring that the network can provide reliable and timely data delivery, while node mobility can impact the connectivity and coverage of the network. Security is also a critical concern, as sensor networks may transmit sensitive or confidential data that needs to be protected from unauthorized access or tampering. Finally, latency is an important consideration for applications that require real-time data, as delays in data delivery can impact the effectiveness of the system.

Figure 3 depicts the sensor device hardware as well as some of its primary components. You can see that the sensor device components are strategically placed on an electric circuit board. Each component is micro in size to cater to mobility. According to (Adamu, 2012), designing effective sensor networks requires careful consideration of these and other factors, as well as the development of innovative solutions to overcome the constraints imposed by the environment and the purpose of the network.

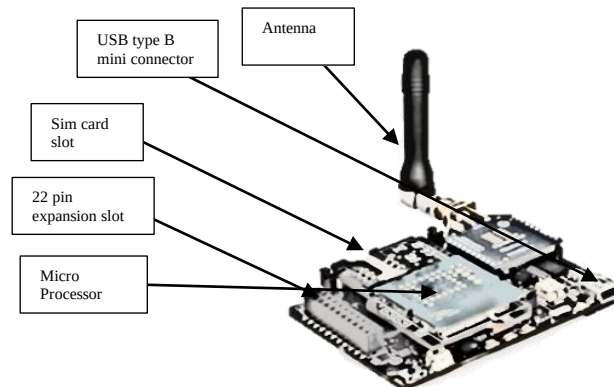


Figure 3: Sensor device used in WSNs Infrastructure for acquiring data and transmitting to sink(Adamu, 2012).

1.0.2 Essence of Quality of Service in WSNs

According to (Mbowe & Oreku, 2014) Quality-of-Service (QoS) is the specific set of service requirements that must be met by the network or application when performing a series of tasks. The support for quality of service in wireless sensor networks (WSNs) has its way of evaluation. As new apps and devices emerge, there is always a need to check the quality of service since it provides assurance and information about the network and software's service level performance. According to (Chen & Varshney, 2004) the QoS requirements for conventional networks mostly give rise to the fame of end-to-end bandwidth furnished in multimedia applications. Most multimedia applications use various measures to ensure that QoS requirements are met, which can be described as QoS parameters or metrics. QoS is always anticipated to be better than the previous service QoS requirements specifications.

Moreover, the two types of QoS which are guaranteed services (Hard QoS) and differentiated services (Soft QoS), for end-to-end users or applications must be met. In the end-to-end QoS support uses a large number of mechanisms and algorithms in different protocol layers while maximizing bandwidth utilization. At the same time, different types of networks may impose specific constraints on QoS support due to their particular characteristics (Chen & Varshney, 2004).

This research used several parameters on soft QoS for end-to-end users/applications to check the overall performance in the original SPIN, GAF, and SPIN-GAF protocols. The block diagram in Figure 4 shows where the QoS meets the needs between the user/application and the network. The QoS model is structured in accordance with the actual implementation. The network sits at the bottom for providing the connectivity between the hardware and software and the QoS requirements are predefined giving instructions on how the system should operate.

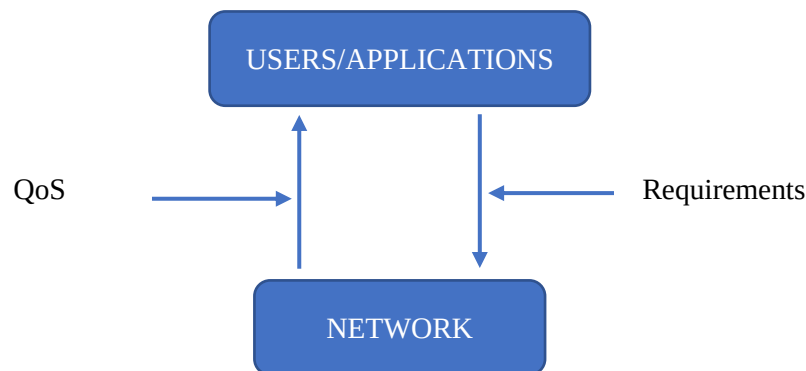


Figure 4: The QoS Model(Mbowe & Oreku, 2014).

1.0.3 Sensor Protocol for Information via Negotiation (SPIN)

The SPIN is a protocol in the data-centric group that uses its dissemination mechanism to broadcast information and monitor the activities within the network (Rehena et al., 2011). Every sensor node in SPIN can communicate in three ways: ADVERTISE (ADV), REQUEST (REQ), and DATA. The first approach is ADV which is used to broadcast the fresh data, after which the sensor nodes issue a REQ to transmit the data, and the DATA is the actual information that is transmitted.

SPIN is activated when a network node receives new data to send out for broadcasting. Figure 5 below depicts the three-way transmission procedure in SPIN. The broadcast mechanism will ADV the data that contain meta-data in a specific message. If any member node has an interest in that particular data, that node will automatically make a REQ for the data, and DATA is the actual message sent to the member node that filed the request. The same procedure will be followed by the remaining member nodes until the full sensor area has received a copy of the data (Sharma et al., 2014).

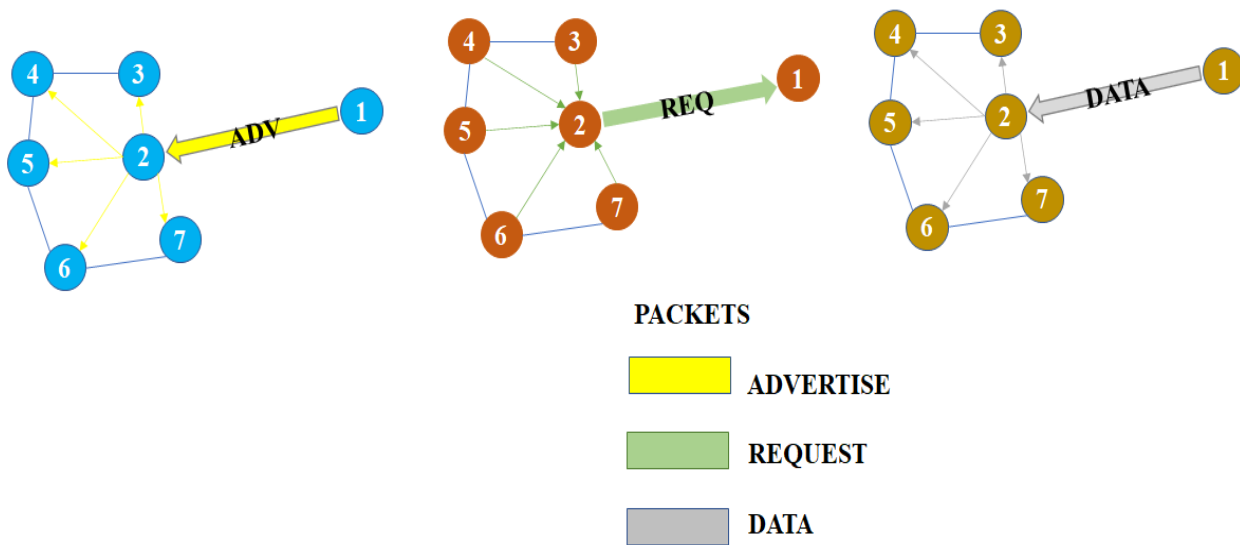


Figure 5: Data transmission patterns in SPIN (Sharma et al., 2014).

1.0.4 Geographic Adaptive Fidelity

Geographic adaptive fidelity (GAF) is a location-based routing protocol that routes data between sensors using the physical position information provided by GPS or other localization systems integrated with sensors (Soni & Mallick, 2015). Geographic Adaptive Fidelity is one of the protocols that serve as a key to identifying the destination of data in the category of location-based routing. Because of such unique performance, the GAF is termed as a power minimization and location identifier routing algorithm intended mainly for mobile networks while it is also possible to implement for wireless sensor networks (Roychowdhury & Patra, 2011).

The network zone in GAF is initially separated according to static regions and established as a virtual grid. Within each region, nodes work together to carry out different tasks. In this case, a group of sensor nodes will select a node with the highest residual energy to be active for a particular amount of time while the rest of the nodes go into sleep mode. The active node takes the task of observing, obtaining, and transmitting data to the sink or base station. In this manner, the energy has been conserved with the help of the sleep mechanism from the GAF and leaves the routing fidelity at a normal state (Walia & Mehta, 2015).

One of the contributions of this research is to formulate an efficient shortest-path algorithm in hybrid SPIN and GAF to find a shorter route to deliver data. As shown in Figure 6, the radios can be killed or turned off to save energy (vital) used during the stationary mode. Sleep state, Discovery state, and Active state are the many motion states used as components of GAF (Walia & Mehta, 2015).

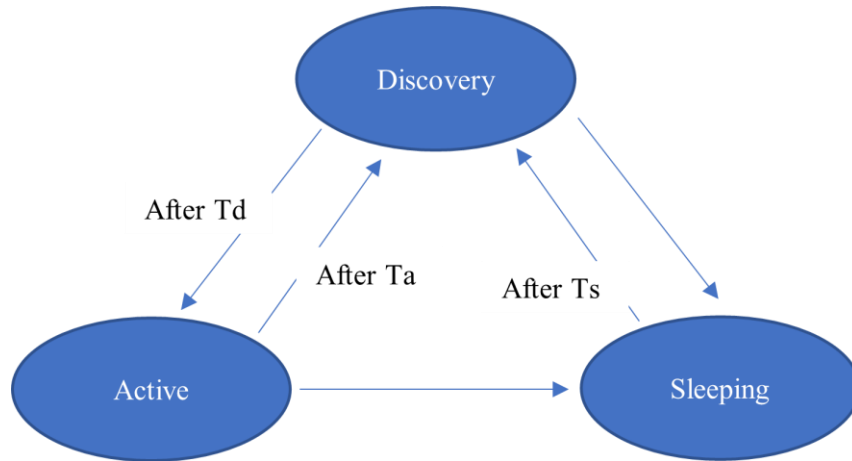


Figure 6: State transition in GAF protocol(Walia & Mehta, 2015).

1.0.5 Shortest Path Algorithm in WSNs

The shortest path algorithm (SPA) is an algorithm that processes paths for connectivity and data transfer in wireless sensor networks (WSNs), facilitating routing among sensor nodes (Khan et al., 2014). The primary function of SPA is to reduce the number of hops before data transmission to a bare minimum. As the shortest path is covered, the shortest path routing algorithms work to conserve energy in WSNs. There are several SPAs, but only a few are applied for WSNs.

Its routing method optimizes two parameters at the same time: energy consumption and delivery rate in networks where sensors are wrongly positioned. Nevertheless, simply optimizing these criteria is insufficient for securing data transmission. To ensure data security, acknowledgment of successful data receipts must be considered (Khan et al., 2014). A new shortest-path algorithm called modified Dijkstra has been devised in this study for the secure routing of data in a wireless sensor network.

1.1 Problem Background and Related Research

As the world moves to digital platforms and the demand for fast communication processes between devices grows in WSN, protocols that perform critical functions become more important. WSNs use networking protocols to build a set of rules that govern how data is sent between network grid devices. Metadata is used in SPIN to transition between sensor nodes. When one node collects a small amount of data, it can make an announcement signal to the other nodes that include the metadata. After receiving the announcement memo, the intended nodes send a transmission requesting the data, while waiting for the data to be provided by the publicist node (Xiao et. al., 2006).

In the SPIN protocol, there is a need to ensure data acquisition and transmission are optimized, because there's no mechanism to ensure packets are forwarded to the destination node or sink for further processing (KalaiMagal & Revathy, 2014).

Dijkstra's shortest path algorithm for wireless sensor networks was invented in 1959 by Dr. Edsger W. Dijkstra, a great Dutch software and computer engineer, who created the famous Dijkstra algorithm in computer science history. Dijkstra's shortest path algorithm begins at the source node and works its way across the graph to find the shortest path connecting all nodes to the ultimate destination. The algorithm will save information for every node in the network about the current available shortest path while checking for another shortest path identified (Umale & Markande, 2015).

If another shortest path is discovered, the new shortest path information will be recorded and marked as visited. The same approach will be repeated until all nodes in the graph have been added to the grid's shortest path. As a result of this operation, we will obtain a path from the source node to every node, showing the shortest way to the target. The disadvantage of Dijkstra's algorithm is that it can only function when the graph has a positive weight; otherwise, the algorithm would fail. It is also not a scalable technique for routing in wireless sensor networks, and it has a small number of stated parameters (Umale & Markande, 2015).

1.2 Problem Statement

The SPIN protocol is a data-centric negotiation-based protocol in WSNs, due to its broadcast architecture, it does not guarantee data delivery. The SPIN protocol makes good use of broadcast channels to deliver a packet to every node in the network, which can result in data vulnerability, transmission delays, or packet loss. The SPIN protocol lacks a data delivery mechanism to ensure smooth data transmission and acknowledgment by the recipient. In this situation, if the node interested in the data is far from the source node and nodes from the source to the destination are uninterested, the data will not be delivered to its destination. The SPIN protocol, which uses the broadcast system for transmission, creates a severe vulnerability and collision in the network, resulting in poor QoS in terms of connectivity and data delivery. The purpose of this research is to solve SPIN's data delivery difficulty by introducing a data delivery mechanism within SPIN (KalaiMagal & Revathy, 2014), (Grover et. al., 2014), (Modieginyane et al., 2018), (Salamath, 2016).

1.3 General Objective

The primary goal of this research is to improve QoS in WSNs by merging the SPIN and GAF location-based algorithms and modifying the existing Dijkstra shortest path algorithm for data delivery in the proposed SPIN-G.

1.3.1 Specific Objectives

1. To develop a protocol that maximizes data delivery by integrating SPIN and GAF location-based algorithms.
2. To implement cluster-based topology for the proposed SPIN-G.
3. To implement the modified Dijkstra shortest path algorithm in SPIN-G.
4. To construct performance metrics for SPIN, GAF, and SPIN-G protocol to check the QoS.
5. To compare and evaluate the performance of the SPIN-G against the SPIN and GAF protocol.

1.3 Research Question

1. How does the integration of SPIN and GAF location-based algorithms in a novel SPIN-G protocol, with the implementation of cluster-based topology and the modified Dijkstra shortest path algorithm, affect data delivery maximization?
2. How does the performance of SPIN-G compare to that of standalone SPIN and GAF protocols, assessed through constructed quality of service (QoS) metrics?"
3. How will the modified Dijkstra algorithm be an effective solution for reducing latency in the proposed SPIN-G protocol?
4. How will the performance metrics evaluate the proposed SPIN-G protocol to ensure QoS is achieved?

1.5 Contribution to Body of Knowledge

Quality of service in wireless sensor networks (WSNs) highlights the behavioral performance of critical components services and applications (Yuan, 2016). The benefits of employing the SPIN protocol are numerous, and the fact that it eliminates flooding and gossiping issues makes it an ideal protocol for WSNs (KalaiMagal & Revathy, 2014). The contribution of this work focused on latency, shortest path, packet delivery rate, signal-to-noise ratio, and throughput to improve the QoS. This study discovered an excellent data delivery mechanism in the GAF protocol after thoroughly examining the protocols for WSNs (Matin & Islam, 2012). The data delivery mechanism in GAF is applied to this research.

Furthermore, this study examined the mode of data dissemination for SPIN and discovered that the broadcast mechanism in the original SPIN was producing delays, collusion, and data integrity issues. Following the merger of the SPIN and GAF data delivery mechanism, this study implemented the clustered grid topology for the upgraded SPIN.

The main contributions of this research to the body of knowledge are:

- Shortest path algorithm (Modified Dijkstra's Algorithm)
- Clustered grid topology (SPIN-G)
- Data delivery mechanism (SPIN-G)

1.6 Research Motivation

This research is based on WSNs and is driven by a plethora of new concepts in the field of computer science. In the field of WSNs, this study and its use of performance metrics to undertake certain QoS evaluations in the SPIN protocol impacted the need to conduct such research. The study focuses on QoS in SPIN because many applications rely on rapid communication services and security for various capabilities. So the research ideally aims to achieve efficient data delivery in the SPIN protocol.

1.7 Thesis Organization

This Thesis comprises five chapters organized in the following order. The first chapter outlines the major components of this research, the research problem, general and specific objectives, research questions, contribution to the body of knowledge, research motivation, and other literature. The second chapter two presents the overview of relevant literature on WSNs, protocols, data delivery mechanisms, and algorithms for identifying the shortest path, etc. Chapter three of this research detailed all the findings and remedies for solving the problem this research is addressing. It contains the methodological approaches taken for each specific objective and simulation tool for implementation, which is meant to analyze the results. The fourth chapter has laid out the results and discussions in detail, while the final chapter five summarizes the entire thesis and explains the future works. Figure 7 below shows how the thesis is structured in a graphic.

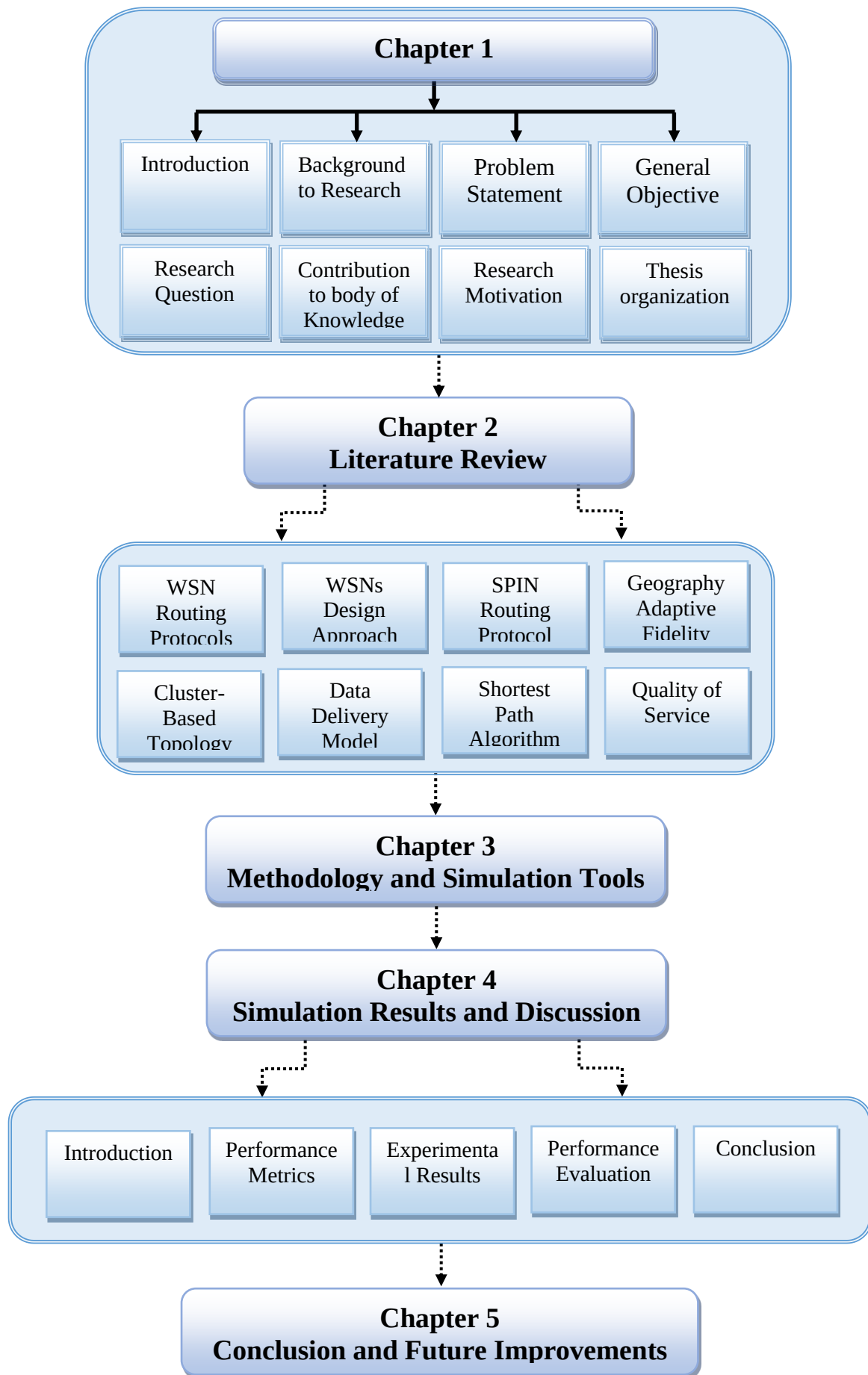


Figure 7: A graphical hierarchical outline of this thesis.

Chapter 2

2.0 Literature Review

Wireless sensor networks (WSN) are addressing various difficulties in traditional network systems in today's environment. As for the existing WSN system, services, and architecture, each topology and protocol has its own set of duties to perform. The SPIN protocol will be thoroughly examined in this chapter to establish why there is no data delivery mechanism inside the SPIN protocol for WSNs. The literature will investigate significant innovations that can improve the SPIN protocol, considering the SPIN protocol doesn't have a mechanism for data delivery. This chapter also reviews a suitable mechanism for data delivery in geography adaptive fidelity (GAF). Further reviews were done to unveil the shortest path algorithm that could improve the quality of service for WSNs. This research discovers that using the GAF location-based method increases the flow of data to the destination or base station in WSNs.

2.1 Wireless Sensor Networks

According to (Matin & Islam, 2012) Wireless Sensor Networks are a customized configuration of wireless sensor devices for a specific basic structureless wireless network to monitor environmental and physical conditions. Some conditions that can be monitored are temperature, sound, pressure, and motion, and relay the information gathered to the BS or sink for data analysis. The sink or BS acts as a link between nodes, operators, and the system, allowing important data from the sensor node to be retrieved by querying the sink for results. A WSN is made up of hundreds or thousands of sensor nodes in strategic positioning. Over the past years, most researchers have contributed to the architecture design, device lifespan, computing proficient algorithm, and WSN protocols, as for the user interaction, a simple application interface is available for monitoring and reporting.

2.2 Wireless Sensor Network Design Approach

The design approach for WSNs is divided into two categories: low-level-based (which prioritizes implementation) and high-level-based (which depends on the model concept to create WSN systems). Each classification approach is detailed on different levels in this literature. The levels are explained from bottom to top, starting with low-level methods. The practice of considering the sensor network as a single system that represents global behavior is known as network-level abstraction, sometimes known as macro programming. The network-level abstraction technique is examined in the form of a database system by (Madden et al., 2005). It creates a platform for users to execute queries written in languages such as SQL. It also focuses on energy efficiency by determining where, when, and how data should be sampled and sent. In this method, the routing tree that distributes any query to receive specified results is also optimized (Madden et al., 2005).

According to the group-level abstraction technique, any node linked to a group-level programming entity is considered a member node for other nodes in the same grid. When nodes are positioned in the same geographical area and have the same number of hops and distance, they are referred to be neighbor nodes, which leads to programming abstractions for WSNs. The technique produces local data processing at neighbor nodes and aids in the construction of distributed algorithms. In Figure 8 the WSNs design approach highlights the two levels of approach to consider in designing WSNs. This level of abstraction includes APIs for finding neighbor nodes, sharing data, and reducing data in its vicinity. In terms of node-level abstraction, a computer programmer can partition a global application into a set of node behaviors capable of running code explicitly on each node. This strategy was largely utilized by operating systems running on virtual machines (BenSaleh et al., 2020).

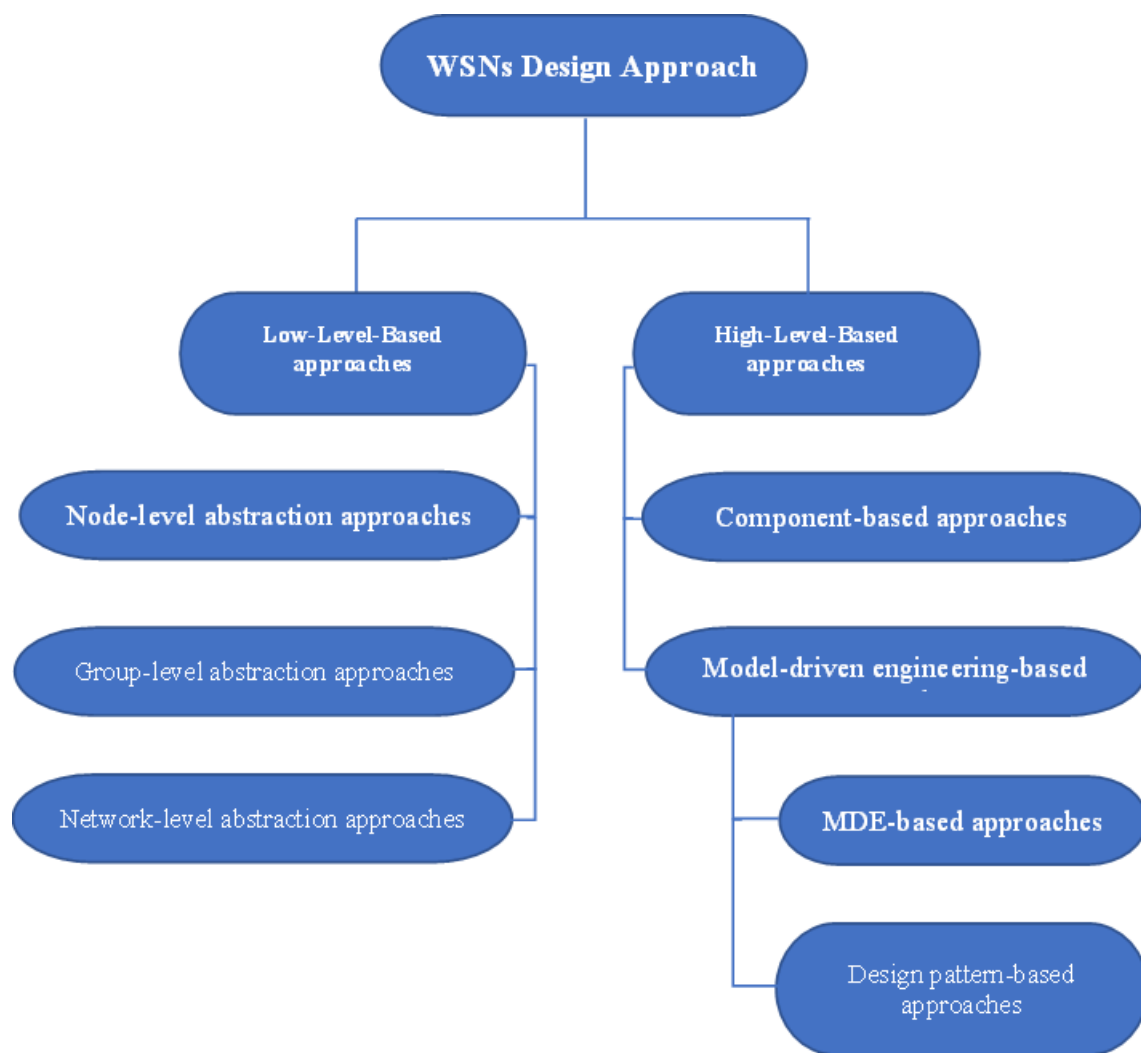


Figure 8: The classification of existing approaches for WSN design (BenSaleh et al., 2020).

2.3 WSN Deployment Technique

According to (Deif & Gadallah, 2014), one of the key layout facets of Wireless Sensor networks is the strategy to deploy sensor nodes. (Deif & Gadallah, 2014) mentioned the two(2) deployment methods which are: random deployment and planned deployment. In a planned deployment technique, the application status of the environment knows that the sensor node can be fixed at the designated region of the application and the operation status of WSN can be fixed. Full coverage of broad environments is attainable in a planned deployment strategy, even where manufactured deployment is challenging.

Random deployment is often referred to as cost-effective deployment. To gain complete coverage of the chosen environment using the random deployment strategy, a large number of nodes should be put in the sensor network. However, this random deployment is only used in situations where full network field coverage is not required. Although sensor deployment can be problematic in some regions, it can be made easier by increasing the coverage effect by installing a high number of sensor nodes to achieve a successful coverage effect of that selected field (Pavithra et al., 2021).

This research chose to deploy the planned deployment method because full coverage of a specific environment is difficult to achieve with the random deployment strategy. The planned deployment was applied in this research with a clustered-based topology to decide the placement of the sensor nodes in the fixed range of two hundred and fifty (250) meters. There are four different mathematical approaches which include artificial potential fields, computational geometry, particle swarm optimization, and genetic algorithms. In this research, a computational geometry-based algorithm was applied to resolve issues of positioning in the SPIN-G protocol.

2.4 Significance of Cluster-Based Topology in WSNs

Sensor nodes are organized into virtual groups for each cluster-based topology. Where clusters are adjacent and adhere to the same rules. Cluster Members, Cluster Gateway, and Cluster Head Node are the properties of a cluster. The cluster head acts as a supervisory node, managing and monitoring the state of the cluster's links and routing data to other clusters in the network. Data is transmitted through cluster gateways in a cluster structure. The cluster members are the nodes that are not the cluster head (Lloret et al., 2009).

The cluster head can regulate the flow of operations and the dimension of a cluster in this network topology, which is usually one to two hops from the cluster head. There are no interconnections between cluster members; only cluster gateways. Two designs are used to classify sensor network clustering techniques which are the cluster head architecture versus the non-clustered head architecture. The cluster head architecture oversees the management and control functions, whereas, in the non-cluster head architecture, any node within the group of nodes can perform the management and control functions.

The cluster head of single-level clustering, on the other hand, is straightforward since it just monitors local topology changes caused by host movement (Lloret et al., 2009). In this study, the non-clustered head design is implemented in SPIN-G, allowing every member node the supervisory responsibility to manage, observe, and route data to other clusters in the network.

Clustered wireless sensor networks are significantly associated with energy conservation, according to (Lloret et al., 2009). In comparison to their non-clustered counterparts, clustered wireless sensor networks have two key advantages: first, they can reduce the amount of inter-node communication by localizing data transmission within the formed clusters and reducing the number of transmissions to the sink node; and second, they can increase node sleep times by allowing cluster heads to coordinate and communicate.

2.5 Importance of Clustering in SPIN

According to (Younis et al., 2006) clustering is a widely used technique in Wireless Sensor Networks (WSNs) for effective data dissemination and network lifetime improvement. The idea is to group sensor nodes into clusters and elect a cluster head (CH) to act as a representative node for the cluster. The CH collects data from member nodes and sends it to the base station (BS) or a sink node. Clustering offers several benefits in WSNs, such as reducing the number of transmissions, minimizing energy consumption, and prolonging network lifetime. By having CHs communicate with the BS, the energy consumption of the member nodes is reduced, as they only need to transmit data to the CH, which then sends the aggregated data to the BS. This helps to conserve energy and prolong the network lifetime.

There are several techniques involved in clustering, including direct communication between member nodes and the CH, as well as transmission of data packets from the CH to the BS. The choice of clustering technique depends on the specific requirements of the application and the network architecture. Overall, clustering is an effective technique for managing energy consumption and prolonging network lifetime in WSNs. It has been used in many WSN protocols and is considered a primary metric for evaluating sensor network performance (Younis et al., 2006).

2.6 Clustered Topology Implemented for SPIN

(Chauhan et al., 2015) talks about the implementation of clustering in SPIN by using the energy-efficient hybrid multi-hop clustering scheme (EEHMCS) protocol. The researchers further mentioned that a new protocol was employed. At the design stage, the new protocol uses two different protocols at different levels: the SPIN protocol within each cluster for data routing to the corresponding cluster head and the EEHMCS protocol at a higher level for data collection by cluster heads and transmission to the base station. The result of this research shows that clustered topology was successfully implemented with the SPIN and EEHMCS forming a spin-based hybrid multi-hop clustering scheme (SHMCS) protocol.

2.7 Clustered Topology Implemented for SPIN-G

SPIN-G (Sensor Protocol for Information via Negotiation Geography) is a wireless sensor network (WSN) protocol that is designed to support distributed event detection and notification and mainly data delivery in WSNs. In SPIN-G, nodes communicate with each other to detect and respond to a given event that might occur in the wireless network. The novel implementation of the SPIN-G protocol this research has designed incorporated the clustered-based topology but without the cluster head in the architectural design. Nodes are organized into clusters, with all nodes acting as the cluster head or coordinator. In this event, all nodes are responsible for collecting data from the receiving nodes, or the nearest node in the cluster and transmitting the data to the sink or based station.

By having the coordinators communicate with only the nodes partaking in the routing process it reduces the amount of network traffic. Second, it can improve energy efficiency by allowing nodes to sleep when they are not needed. Since the coordinator is responsible for collecting and processing data, the other nodes can sleep until they are needed to collect data. This can extend the battery life of the nodes in the network. Another benefit is that it can reduce latency in the network. Since all nodes serve as coordinators, it creates the opportunity for efficient routing within the clustered grid. In the original clustered topology, if the coordinator node fails or becomes overloaded, it can affect the entire cluster. The non-clustered head topology is the solution to that issue.

2.8 Non-Clustered Head Architecture in SPIN-G

In this research, the non-clustered head architecture was implemented for the proposed sensor protocol for information via negotiation geography (SPIN-G). The importance of the non-clustered head architecture is to give all sensor nodes the ability to partake in the routing process. Another key reason is simply that we want the shortest path algorithm to find the shortest path using every nearest available node possible. One way this research discovers is by allowing all available nodes to participate in the routing process if the shortest path algorithm finds its fit as a node in the nearest range to the destination or the BS. This approach could potentially have some advantages over traditional clustered-based topologies with a cluster head. For example, it may reduce the overhead associated with cluster head selection and communication, as all nodes can participate in the routing process.

2.9 Data Delivery Model for WSN

In WSN, the process of information capturing and dissemination can be accomplished in a variety of ways. The data delivery models in Figure 9 define the movement of data from receiving sensor nodes to the sink or base station (BS). The same paradigm for delivering data tells us that it is classified as event-driven, query-driven, continuous, or hybrid. In the event case, sensor nodes perform data acquisition when an event of interest occurs (Sahar et al., 2021).

In the case of query-driven, the method is to force nodes to wait and get information about the sensed data meant for them. In continuous mode, sensor nodes broadcast at regular intervals, transmitting information regarding perceived data at a predetermined rate. At last, the hybrid is a blend of the data delivery models, so by combining the models, sensor nodes will broadcast data at regular intervals while also responding to questions. One thing they all have in common is that they are all programmed to provide information about interesting happenings (Sahar et al., 2021).

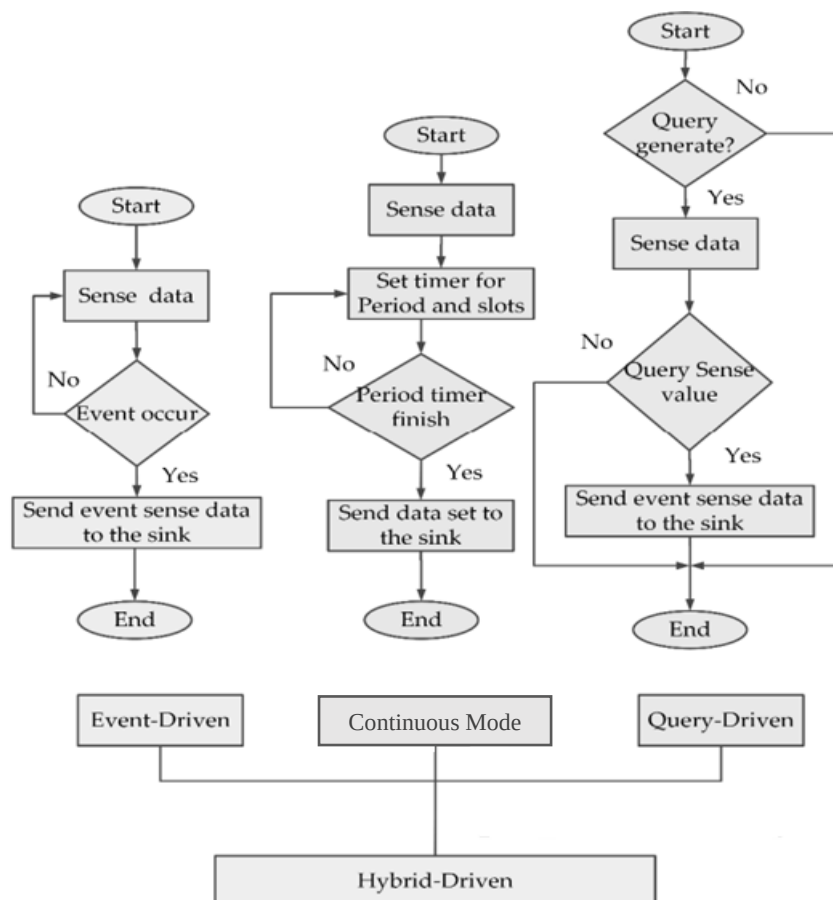


Figure 9: The flowchart diagram shows the four data-driven model states in WSN(Sahar et al., 2021).

2.10 Data Transmission Issues in SPIN

The SPIN protocol falls under the data-centric routing group that uses the broadcast mechanism for the transmission of data (Kaur et al., 2013). The SPIN protocol was implemented for WSNs and one of its functions is to eliminate the problems parallel to gossiping and flooding that could happen in some protocols. The key logic is distributing the data, which is acquired by the sensor node, it could consume a lot of resources like metadata that serves as stored information relating to sensed data. Those resources can be managed by the resource manager in every node and adjust their functionality accordingly to maintain some level of consistency in the protocol.

Figure 10 shows the operation principle used in the existing SPIN protocol for data exchange. Metadata is the name given to the medium used for data exchange in SPIN. It is a fixed data description that provides information on a specific data acquisition. The metadata is then distributed to other sensor nodes. The node that obtains the data broadcasts a broadcast message containing the metadata to all nodes in the network. While it waits for the destination node to send a request message for the data to be transmitted to the requested destination after receiving the broadcast message (Kaur et al., 2013).

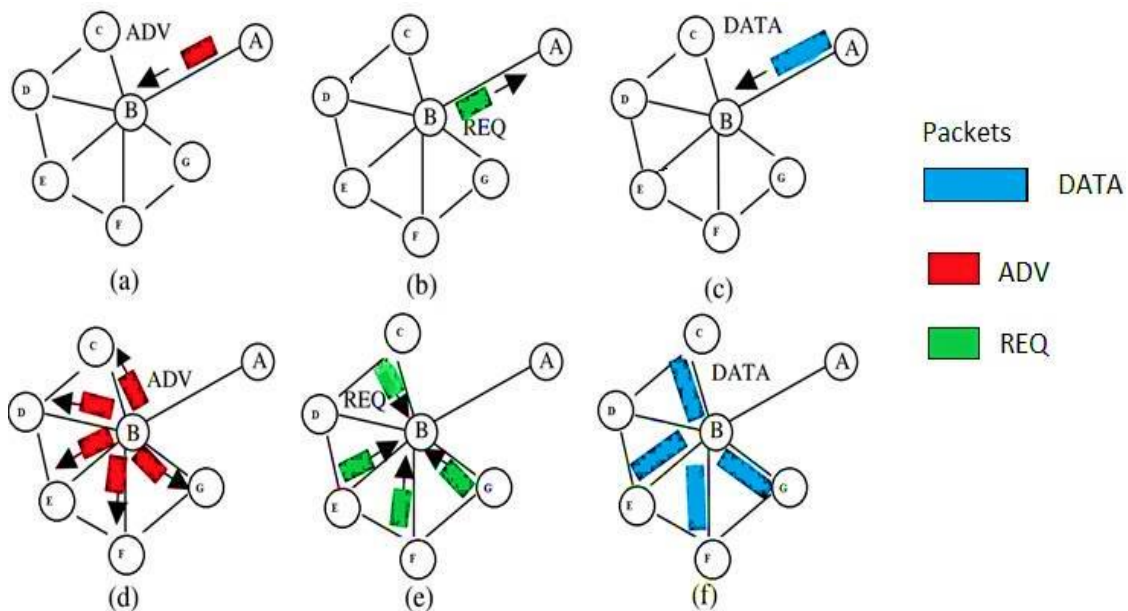


Figure 10: Three operation principles in SPIN using the broadcast mechanism

According to (Salamath, 2016) the dissemination of data in the network through the SPIN protocol takes a longer time. The node with much more computation consumes more power. While fewer sensor nodes may be used several times and those nodes may lose energy earlier than other nodes in the network.

SPIN is based on data-centric routing, with sensor nodes broadcasting ADV messages for the data they have and waiting for REQ messages from interested sinks or nodes. SPIN does not support the semantics of its meta-data format because it is application-dependent. In other words, SPIN names sensed data using application-specific meta-data. Although SPIN has some advantages, such as: (1) resolving the issues relating to classic flooding protocols and (2) localizing topological changes, it also has some disadvantages, such as (1) scalability, SPIN is not scalable, (2) if the sink is interested in several events, the sensor nodes nearby may drain their energy, and (3) SPIN's data advertisement technique cannot guarantee data delivery if the interested nodes are far away from the source node (R. KalaiMagal & M. Revathy, 2014).

2.11 Data Delivery in SPIN-G

The data delivery process in SPIN-G is initiated by the query message or data acquired by the sensor node, just as it is in the original SPIN. The data contains a set of parameters that indicate the data's interest before transmission by the nearest available node. The query message contains information such as the type of data, sensor node location, and data collection period. SPIN-G, a location-based algorithm, will obtain metadata containing destination information for specific data to prepare for transmission. The enhanced Dijkstra shortest path algorithm will then calculate all possible routes to the destination or sink to determine the shortest transmission path.

The sensor nodes that acquire the query message use the parameters to filter the data and send a response message back to the sink node with the relevant data. The response message may contain the raw data or a summary of the data, depending on the query parameters. To optimize the data delivery process, SPIN-G uses a location-based approach that focuses on the shortest available route rather than the sensor nodes that produce it. This approach enables efficient data dissemination, as sensor nodes can transmit only the relevant data at a low latency rate to the sink node, thereby conserving energy and reducing network congestion.

2.12 Essence of Geography Adaptive Fidelity in WSN

GAF is a routing protocol that is based on location. It improves the routing protocol in terms of data distribution, energy efficiency, and sensor network self-configuration. Position-based routing ensures that the data is transmitted from the source node to the destination node using its location information. One sensor node chooses the path to take to send data using this method based on the location of the destination, sink, and sensor nodes in the grid. Many well-known WSN protocols do not include location information, resulting in high energy consumption (Walia & Mehta, 2015).

The location-based algorithms use information about the location of data to help facilitate a smooth and reliable route for transmission. Nodes that are close to each other or in the same grid can share position data. Data sharing helps to ensure that location information is sent to all intended network nodes before a packet is forwarded. The data-sharing process makes it simple to acquire data and includes the destination coordinates in the packet before transmission. Geographical algorithms can obtain location information from a specific system, such as GPS, or virtual coordinates. Geo-casting is a method of delivering data to sensor nodes in a network grid using geographical protocols.

This method is considered appropriate because, in WSN, the sink or BS normally requests information from network grid sensor nodes. The region is created dynamically by communicating nodes, or it can be statically defined by the source node to prevent other sensor nodes from diverging. Furthermore, geographical positioning can be used as a communication area for nodes transmitting and receiving data in location-based routing mechanisms. Location-based communication region mechanisms can be an extremely effective way to locate services or resources. This study sees the location-based routing mechanism as a smart mechanism for implementation in SPIN as a means of resolving data delivery issues within the protocol (Walia & Mehta, 2015).

2.13 Shortest Path Algorithm and Drawbacks

Dijkstra was the first to develop the shortest path algorithm in, 1959. Dijkstra's algorithm is intended for routing, specifically tracking, and takes several parameters into account, including the number of nodes, throughput, packet delivery ratio, and energy optimization in WSN. Dijkstra's algorithm is intended for pointless graphs. Dijkstra's Algorithm is the parent algorithm for several other shortest-path algorithms. Several shortest path algorithms have been discovered to date, some of which are applicable for routing in WSNs and others are not. Dijkstra's Algorithm is used to find the shortest path in an undirected graph. The disadvantage of Dijkstra's algorithm is that it can only function when the graph has a positive weight; otherwise, the algorithm will fail. It is also not a scalable algorithm for routing in wireless sensor networks, and it has a small number of defined parameters. (Khan et al., 2014).

However, another algorithm for the shortest route was calculated and intended for a single source. Floyd developed FloydWarshall's algorithm for every pair of shortest paths, and Warshall developed the outcome-based formula in 1962. Other researchers have recently investigated solutions for the shortest path to combat some of the issues associated with routing traffic in network communications and vehicle routing in transmission. Even though Floyd Warshall's algorithm is meant for calculating the shortest path, it is not suitable for implementation in a wireless sensor network. It lacks scalability and efficiency and can only work with a limited number of parameters. Nonetheless, the enhanced Floyd-Warshall SPA conducts a discussion on optimizing energy for WSN. Modified FloydWarshall's algorithm is implemented for a random topology system (Khan et al., 2014).

2.14 Dijkstra Algorithm

Dijkstra's algorithm is a popular algorithm used for finding the shortest path between two nodes in a graph with non-negative edge weights. The algorithm works by maintaining a set of "visited" nodes and a set of "unvisited" nodes. Initially, all nodes are in the "unvisited" set. The algorithm then starts at the source node and explores its neighbors, updating their distances from the source if a shorter path is found. This process continues until the destination node is visited or all nodes in the "unvisited" set have been explored (Khan et al., 2014).

Dijkstra's Algorithm is stated as follows:

1. Enter the input graph, number of vertices n , source vertex vs and destination vertex vd as parameters in a function, say $dijk(a[][] , n, vs, vd)$.
2. Repeat Step3 for $i=1, 2, \dots, n$.
Reset array element $set[i]$ to 0
3. Repeat loop for $i=1, 2, \dots, n$.
Check if $a[vs, i]=0$ [In case no direct edge between vs and vi].
Then: $leng[i]=Infinity$ [Infinity value is taken to be 999] and $path[i]=0$ [Invalid path].
Else: $leng[i]=a[vs, i]$ and $path[i]=vs$ [vs is immediate predecessor of vi].
[End of loop.]
4. Set the value $set[vs]$ to 1 and reset the value $leng[vs]$ to 0.
5. Repeat loop while(vd is not included in the $set[]$). Set $j=search_min(leng,set,n)$ [This function will return a vertex with a minimum label such that it is not included in $set[]$ and value of $set[j]=1$].
6. Repeat loop for $i=1, 2, \dots, n$.
Check if vi is there or not in $set []$, then check if vi and vj are connected by an edge. If yes, then check if $leng[j]+a[i,j]<leng[i]$ [Replacement occurs if the existing label is not minimum]. If yes, then $leng[i]=leng[j]+a[i,j]$ and $path[i]=j$
[End of loop.]
[End of Step5 loop.]
7. Set $weight=leng[vd]$.
8. Print the shortest path weight and then the shortest path by backtracking.
9. Exit.

2.15 Modified Dijkstra Algorithm

Modified Dijkstra's Algorithm is an extension of the classical Dijkstra's algorithm for finding the shortest path in a weighted graph. The modification allows the algorithm to handle negative edge weights, making it applicable to graphs with negative-weight edges but without negative cycles. This modified version is also known as "Dijkstra's Algorithm with Negative Edge Weights" or "Dijkstra's Algorithm with a Minimum Heap" (Natarajan & Rajendran, 2014).

Here's the pseudo-code for the Modified Dijkstra's Algorithm:

```
procedure ModifiedDijkstra(Graph, source):
    create a priority queue Q
    create a distance array dist[] of size V (number of vertices) and initialize all distances to
    infinity
    create a boolean array visited[] of size V and initialize all elements to false

    dist[source] = 0
    Q.push((source, 0))

    while Q is not empty:
        u, dist_u = Q.pop() // Extract the vertex with the smallest distance from the priority
        queue

        if visited[u]:
            continue

        visited[u] = true

        for each neighbor v of u:
            // Calculate the new distance from source to v passing through u
            alt = dist_u + weight(u, v) // weight(u, v) gives the weight of the edge between u and
            v

            if alt < dist[v]:
                dist[v] = alt
                Q.push((v, alt))

    return dist[]
```

ALGORITHM I – Modified Dijkstra's Algorithm to compute n paths in G from S to D

Input: Source node S , Destination node D , Graph (Ad hoc network) G , Number of paths n , Cost function f_c , Group of vertices g_v , and Group of arcs g_a .

Output: Multiple paths $P_1, P_2 \dots P_n$

begin

$f_{c_1} = f_c$ // Initial f_c
 $G_1 = G$ // Initial G

for $x = 1$ to n **do**

$T_{s_x} = \text{Dijkstra}(G_x, S)$

$P_x = \text{Get_Path}(T_{s_x}, D)$

for each arc a in g_a

$t_{l_a} = (1 - W_l) \cdot t_{l_a}$

end for

for each vertex v in g_v

$t_{l_v} = (1 - W_n) \cdot t_{l_v}$

end for

for each arc a in g_a

if a or a^{-1} is in P_x

$f_{c_x+1} = f_c(f_{c_x}(a))$

else if $\text{Vertex_Head}(a)$ is in P_x

$f_{c_x+1} = f_a(f_{c_x}(a))$

else

$f_{c_x+1} = f_{c_x}$

end if-else

end for

$G_{x+1} = (g_v, g_a, f_{c_x+1})$

end for

return $(P_1, P_2 \dots P_n)$

end

2.16 Quality of Service Necessities in Wireless Sensor Networks

In wireless networks, quality of service refers to the amount of service provided to network operators. End-to-end delay, throughput, packet loss probability, signal noise ratio, and other services are among those provided. QoS in a network is primarily determined by the service provider, who sets the service requirements based on the specific standard and organizational needs; it may or may not be a universal standard (Service Level Agreement). Most service requirements take into account both the network user's and the service provider's needs (Mbowe & Oreku, 2014).

Wireless sensor networks (WSNs) are emerging as the foundational infrastructure for new classes of large-scale networked embedded systems. However, WSN system designers must meet the applications' quality-of-service (QoS) requirements and user requirements specifications. Extremely harsh and dynamic physical environments, as well as extremely limited energy, computing, memory, and communication node resources, present significant challenges to meeting QoS metrics such as reliability, timeliness, and system lifetime. The "hidden node problem" results from the limited communication range of WSN nodes, link asymmetry, and physical environment characteristics, and it is a major source of QoS degradation in WSNs (Koubaa et al., 2009).

2.17 Quality of Service Challenges in Wireless Sensor Networks

(Sharma et al., 2013) highlights some of the challenges involved in ensuring Quality of Service (QoS) in Wireless Sensor Networks (WSNs). WSNs are composed of a large number of sensor nodes that collect data and transmit it to a sink node for processing. One of the major challenges in routing data in WSNs is the inaccurate state information that is available. This can make it difficult to balance the traffic in the network, leading to imbalanced QoS-restricted traffic. To ensure a minimum required QoS, it is necessary to allocate sufficient bandwidth to the sensor network. However, designing QoS techniques for unbalanced traffic can be challenging. In addition, routing in WSNs often has to sacrifice energy efficiency to ensure packet delivery. While multi-hop data collection can be more energy-efficient overall, the overhead can slow down packet delivery.

Another important consideration in WSNs is the addition or removal of nodes. According to Sharma et al. (2013), these events should not have an impact on the network's QoS. Overall, ensuring QoS in WSNs is a complex task that requires careful consideration of various factors, including energy efficiency, traffic balancing, and node management (Sharma et al.,2013).

Sensor networks are indeed typically composed of a large number of nodes, often in the hundreds of thousands or more, and the cost of each node is a critical factor in the design of such networks. Flash memory is often used in sensor networks because of its low cost and non-volatile nature. The choice of the central processing unit (CPU) is also important for sensor nodes, as it affects both the energy consumption and computational power of the node. There are various options for CPU implementation, including microcontrollers, microprocessors, and field programmable gate arrays (FPGAs).

Microcontrollers are often a good choice for sensor nodes because they can be designed to have three states: active, sleep, and idle. This allows for effective energy management, as the node can be put into a low-power sleep state when it is not actively processing data. FPGAs, on the other hand, are less suitable for power-constrained applications because their energy consumption cannot be reduced, and it is not possible to create separate blocks for different functions to optimize power usage. While FPGAs can be useful for some applications that require high performance and flexibility, they are generally not recommended for use in sensor networks where power consumption is a critical factor. Overall, the choice of CPU for a sensor node depends on various factors, including the specific requirements of the application, the power budget, and the desired performance characteristics. The sensor network and sensor node architecture have an important role to play in the QoS according to (Sharma et al.,2013).

In wireless sensor networks (WSNs), the sensor nodes need to have a large radio range to establish network connectivity and collect data effectively. This is particularly important in environments where there is no existing communication infrastructure, such as in remote or rural areas. To ensure that WSNs are cost-effective, it's important to develop software that is lightweight and energy-efficient, as well as hardware-agnostic. This means that the software should be able to run on different types of hardware so that users have more flexibility in choosing the hardware that best suits their needs.

To reduce energy consumption and extend the lifespan of the sensor nodes, it's important to develop algorithms and protocols that are less complex. Complex algorithms and protocols require more processing power, which in turn requires more energy. By developing simpler algorithms and protocols, it's possible to reduce the energy consumption of the sensor nodes and extend their lifespan, which is crucial in many WSN applications (Sharma et al.,2013).

2.18 Quality of Service in SPIN

(Patel & Raja, 2014) highlighted some routing protocols in Table 1 below to show which protocol supports quality of service (QoS) in WSNs. SPIN is one of the eight (8) routing protocols on the list with no support for QoS. However, SPIN does not explicitly support QoS, and its performance in terms of QoS may be affected by factors such as network topology, traffic patterns, and node mobility.

Table 1: Comparison of QoS-aware routing protocols in WSN (Patel & Raja, 2014).

Routing Protocols	Routing Structure	Scalability	Mobility	Network Lifetime	QoS	Data Aggregation	Multipath
SPIN	Flat	Limited	No	Good	No	Yes	Yes
SAR	QoS	Limited	Yes	Good	Yes	Yes	Yes
SPEED	QoS	Limited	Yes	Good	Yes	No	No
PEGASIS	Hierarchical	Good	Yes	Very Good	Yes	Yes	No
GEAR	Location	Limited	No	Very Good	Yes	No	No
DD	Flat	Limited	No	Good	No	Yes	Yes
LEACH	Hierarchical	Good	Yes	Very Good	Yes	Yes	No
TEEN &APTEEN	Hierarchical	Good	Yes	Very Good	Yes	Yes	No

2.19 Quality of Service of SPIN-G

The SPIN-G protocol's quality of service (QoS) is primarily concerned with data delivery efficiency. This research took into account all aspects of data transmission while implementing the SPIN-G. SPIN-G is designed to eliminate data delivery fraud and transmission delays that plagued the original SPIN protocol. In SPIN-G techniques such as data aggregation, adaptive routing, and traffic differentiation provide QoS guarantees, similar to ordinary SPIN. Data aggregation is the process of combining data from various sensors to reduce redundant transmission and deliver fused information to the base station (Dehkordi et al., 2019). Data aggregation typically involves combining data from various sensors at intermediate nodes and sending the combined data to the base station (sink). Data aggregation acquires crucial data from sensors and transmits it to the base station with minimum latency (Dehkordi et al., 2019).

To facilitate routing, the SPIN-G deployed the clustered grid topology without the cluster head in the adaptive routing technique. In terms of traffic differentiation, the Dijkstra shortest path algorithm will calculate the shortest path from the source node to the base station to reduce network traffic congestion. This study analyzed the performance of SPIN-G using performance metrics for testing QoS.

2.20 Performance Metrics for WSNs

Designing wireless sensor networks for sustainable data collection involves creating efficient and environmentally conscious systems that optimize the collection and transmission of data. The practical issues that are important to consider when designing wireless sensor networks for various applications. One of the advantages they highlight is the lower installation and maintenance costs associated with wireless sensors and controls compared to wired systems. Wireless sensors can be installed and configured without the need for complex wiring, which can be expensive and time-consuming. Additionally, wireless sensors do not require regular maintenance and inspections that are necessary for wired systems, which can save time and resources. Overall, wireless sensor networks offer numerous benefits for various applications, including lower installation and maintenance costs, increased flexibility, and improved data collection and analysis capabilities (Jang & Healy, 2009).

(Alchihabi et al., 2018) mentioned a few metrics that are popularly used to evaluate the performance of WSNs. As for the simulation setup and operating conditions in WSNs, parameters are employed to consistently check the real-time network conditions that may occur at a given event.

In "Performance Evaluation Metrics for Wireless Sensor Networks: A Survey", (Alchihabi et al., 2018) discuss various metrics that are commonly used to evaluate the performance of Wireless Sensor Networks (WSNs). These metrics can be used to assess different aspects of WSNs, such as network lifetime, energy efficiency, reliability, throughput, and latency. In addition to performance metrics, the authors highlight the importance of simulation setup and operating conditions in WSNs. Simulation parameters are used to consistently check the real-time network conditions that may occur at a given event.

These conditions can include the number of nodes in the network, the type of applications being used, the distance between nodes, and the transmission power of each node, among others. By using simulation parameters to evaluate WSNs under various conditions, researchers can better understand the behavior of the network and identify potential performance issues. This information can be used to optimize the design and configuration of WSNs to improve their performance and efficiency as well as the quality of the service (QoS).

2.21 Throughput in WSN

Throughput is an essential performance metric in wireless sensor networks (WSNs). It measures the amount of data that can be transmitted through the network in a given amount of time, while taking into account factors such as congestion, packet loss, and delay. Even though all receiving nodes in the network have limited queue sizes, it calculates the traffic infused per unit of time from all of them. A higher amount of throughput is desirable in WSNs because it enables the efficient and timely delivery of data from the sensor nodes to the base station (BS). As mentioned, the throughput in WSNs is affected by various factors such as network topology, routing protocol, congestion control, and energy efficiency mechanisms. Therefore, it is important to design and optimize these factors to achieve better performance in terms of throughput and other relevant metrics (Kadu & Deshpande, 2012).

Throughput refers to the rate at which messages are sent or the amount of data that can be transmitted over a communication channel within a given period. It is usually measured in bits per second (bps) or bytes per second (Bps). In a wireless sensor network (WSN), increasing throughput requires the use of a high number of packet sizes. However, as the base station load increases, congestion and delays may occur, leading to a decrease in throughput (Kadu & Deshpande, 2012).

2.22 Throughput in SPIN-G

In SPIN-G, throughput is used to find the average amount of data transmitted in a given period of time. Throughput is reliant on various factors, such as the data rate, packet size, signal quality, and network topology. Generally, the data rate and packet size determine the maximum throughput that can be achieved by the SPIN-G with the aid of clustered grid topology. The SPIN-G protocol is a hybrid protocol that combines the location-based algorithm in GAF with the SPIN protocol. This combination is expected to result in an optimal rate of throughput, making SPIN-G one of the best WSN data transfer protocols. This research implemented the following in SPIN-G: a clustered grid topology, the Dijkstra shortest path algorithm, and the GAF location-based algorithm. The purpose of implementing those is to achieve a maximum data delivery rate in the SPIN-G. Other factors that can contribute to improving the throughput in SPIN-G are:

1. Network Topology Optimization:

This can significantly impact the throughput of the SPIN-G protocol. Optimizing the SPIN-G topology by reducing the distance or range between sensor nodes can automatically boost the throughput for SPIN-G.

2. Signal Quality:

The quality of the signal covers the strength, clarity, and stability of signals being transmitted or received through a communication channel. In the context of electronic devices, it is usually associated with the quality of audio, video, or data transmission. Increasing the signal quality makes SPIN-G free from interference, distortion, or noise, and can high throughput with minimal error or loss.

3. Increase Data Rate :

Upgrading pieces of networking equipment in the infrastructure to a higher version. Router, modem, and network interface card are some of the devices that can increase the data rate.

4. Increase Bandwidth:

The bandwidth plays an important role in achieving higher throughput in SPIN-G. Increasing bandwidth speeds up the internet connection, which can lead to increased throughput in SPIN-G.

2.23 Overall Summary of Literature

The survey of literature in this chapter provides a comprehensive overview of the challenge this research addresses. The review shows that indeed there's no such mechanism for data delivery in SPIN leaving a drawback in terms of its throughput. The focal point is the data delivery issue and the appropriate solution to eliminate the problem. This study delves into the SPIN protocol and proposes a vibrant solution that makes use of the Geopraghy Adaptive Fedility location-based algorithm to predefine the routes to the destination for all data packets to be delivered. The location-based mechanism was integrated into SPIN, resulting in the novel SPIN-G protocol. Considering the need for a shortest path algorithm in SPIN-G, this research examines the Dijkstra shortest path algorithm.

Chapter 3

3.0 Methodology

The ideal methods used in this study are fundamentally based on each objective being applied using a relevant method, conceptual modeling, but with a particular emphasis on the qualitative research method, which led to the whys and hows of the phenomenon that ensures QoS for WSNs. This novel study's data is entirely research-based, with implementation and experiments carried out using a MATLAB simulator. Several previous research papers were reviewed to facilitate the literature of this study, which was critical to the process, particularly information from other contributors. A research approach was applied to the SPIN protocol through rigorous checks to identify the protocol's shortcomings and potential solutions, revealing that the SPIN protocol lacks a data delivery mechanism and topology. To highlight the findings in this chapter, the following methodological approaches were used:

3.0.1 To develop a protocol that maximizes data delivery by integrating SPIN and GAF location-based algorithms.

In this study, the SPIN and GAF location algorithm integration is simulated using MATLAB, and all the results are interpreted in the result and analysis column of this thesis. These findings are presented in various graphs and tables for analytic and comparison purposes, and they clearly demonstrate the throughput of SPIN, GAF, and the proposed SPIN-G protocol. The parameters of packet loss and packet delivery ratio were tested. The data delivery mechanism, according to GAF, makes use of highly sensitive data location information.

The following are the steps taken in this study to integrate the SPIN and GAF location algorithms with MATLAB:

- I. Get the SPIN and GAF protocol software from their respective websites or repositories.
- II. Install the required dependencies for the SPIN and GAF protocols to function on your operating system.

- III. Using the add path command, add the SPIN and GAF protocol directories to the MATLAB path.
- IV. Using the load command, load the sensor data you want to analyze into MATLAB.
- V. Preprocess the data in accordance with the SPIN and GAF protocols. Filtering, calibration, and other preprocessing steps may be included.
- VI. Use the SPIN protocol to negotiate data transmission between network sensors. This could entail utilizing the SPIN protocol functions included in the software package.
- VII. Use the GAF protocol to adaptively adjust the fidelity of the transmitted data based on the sensor network's geography. This could entail utilizing the GAF protocol functions included in the software package.
- VIII. Analyze the data from the sensor network using MATLAB functions and algorithms to extract useful information and insights.

The SPIN-G will perform a random check to initialize the position of all cluster members, and then map the cluster-based topology in the network grid based on the results. After all of the cluster nodes have been mapped into the grid, each grid/clustered member will take on a supervisory role to ensure data delivery. The active sensor nodes in each network grid will use the location information to deliver the data if interested or not. Overall, integrating SPIN and GAF protocols with MATLAB provides this research with a powerful toolkit for analyzing sensor network data specifically the proposed SPIN-G protocol, and extracting valuable insights from it.

3.0.2 To implement cluster-based topology for the proposed SPIN-G

To implement cluster-based topology for the proposed SPIN-G in MATLAB, this research follows the steps outlined below:

- I. Define the network topology: This research implemented the clustered grid topology, with the use of a non-clustered head. For a cluster-based topology, this research divides the network into smaller clusters of sensors that can communicate with each other in the grid. This research used graph theory and network analysis tools in MATLAB to model and visualize the network topology.
- II. Implement the SPIN-G protocol: In MATLAB, implement the SPIN-G protocol to enable reliable data delivery and information exchange between network sensors. Defining the negotiation protocol, determining the shortest packet transmission route, designing the message format, and implementing the data delivery exchange mechanism are all part of this process.
- III. Implement the clustering algorithm: In MATLAB, choose a suitable clustering algorithm for your network topology, such as k-means or hierarchical clustering. This study used a hierarchical clustering algorithm to group sensors based on their location, connectivity, or other criteria.
- IV. Implement inter-cluster communication: Because this is a non-clustered head architecture, implement a communication protocol between cluster members to facilitate inter-cluster communication. This process entails creating a message format, defining communication channels, and putting the data delivery mechanism in place.
- V. Implementation testing and refinement: Test the SPIN-G cluster-based topology implementation in MATLAB with simulated or real-world sensor data. Refine the implementation as needed to ensure data delivery in SPIN-G by optimizing performance, scalability, and reliability.

The proposed SPIN-G protocol's cluster topology was implemented using the MATLAB simulator. The cluster topology in this study includes a non-cluster head architecture and places all nodes in strategic areas across the grid to perform supervisory duties. In the cluster topology, the on-grid is used to create an interconnection between cluster members. Following that, the results will be displayed in a graph that analyzes the cluster-based topology between SPIN, GAF, and the proposed SPIN-G protocol. Overall, implementing the proposed SPIN-G protocol's cluster-based topology in MATLAB can enable efficient and dependable communication and information exchange between sensors in a large sensor network.

3.0.3 To implement the modified Dijkstra shortest path algorithm in SPIN-G

To implement the modified Dijkstra shortest path algorithm in SPIN-G, you can follow these steps:

- I. Understand the modified Dijkstra algorithm: The modified Dijkstra algorithm is a graph-finding algorithm that finds the shortest path between two vertices. It works by selecting the vertex with the shortest distance from the source and updating the distances of its adjacent vertices iteratively.
- II. Obtain an existing modified Dijkstra algorithm implementation: This research obtains the existing modified Dijkstra algorithm MATLAB source code implementations. The modified Dijkstra MATLAB source code was imported into MATLAB, and these implementations were used as a starting point for customizing the algorithm to suit the specifications of SPIN-G.
- III. Recognize the data structures used in the current implementation: The modified Dijkstra algorithm typically stores vertices based on their distance from the source vertex in a priority queue. This study identified the need to understand how the priority queue is implemented in existing code, as well as how distances and previous vertices are stored. This study was able to interpret the data structure used by understanding the priority queue concept.

- IV. The modified Dijkstra algorithm to meet your needs: Depending on your specific needs, you may be required to modify the existing Dijkstra algorithm implementation. For example, this study identified the need to add constraints or heuristics to the algorithm to meet the SPIN-G requirements.
- V. Test the modified algorithm: Once the Dijkstra algorithm has been modified, it should be thoroughly tested to ensure that it produces the best and most desirable results. Sample graphs can be used to test the modified Dijkstra algorithm and compare its output to the expected results in MATLAB.
- VI. Integrate the modified Dijkstra algorithm into your project: After a thorough test and proven to be satisfactory with the modified Dijkstra algorithm's performance, this study incorporated it into the SPIN-G project. This research ensures that the modified Dijkstra algorithm is compatible with the rest of the code and does not introduce any new bugs or problems.

The modified Dijkstra shortest path algorithm (SPA) will calculate the number of hoops in each network grid from the source to the final destination. So, a time latency formula is implemented into the proposed SPIN-G which is meant to calculate the number of hops from source to destination to determine the shortest path. The program criterion will be based on the minimum number of hoops, if the lowest hoops count is e.g., fifteen (15) then it will be the shortest path.

- Check the location information of the data.
- Dijkstra SPA counts the number of hoops and selects the minimum hoop count.
- All active nodes within the path ensure the data's safe passage to the final destination.

Afterward, the results are shown in a graph analyzing the latency between SPIN and the proposed SPIN-G protocol. Testing the following two parameters, end-to-end delay, and signal-to-noise ratio.

3.0.4 To construct performance metrics for SPIN, GAF, and SPIN-G protocol to check the QoS

This study first identifies the specific QoS attributes that we want to measure to design performance metrics for SPIN, GAF, and SPIN-G to check the Quality of Service (QoS). The following are some common QoS characteristics that can be used to evaluate those protocols:

- I. Reliability: the ability of the system to provide consistent and dependable service.
- II. Throughput: the amount of data that can be transmitted by the system per unit of time.
- III. Latency: the time it takes for a request to be processed by the system.
- IV. Scalability: the ability of the system to handle an increasing number of users or requests.

Based on these QoS attributes, we can create the following performance metrics for SPIN, GAF, and SPIN-G protocols:

- I. Reliability: The system's reliability can be calculated by calculating the percentage of requests that are successfully processed without errors or failures. The better the system, the lower the error rate (bit error rate).
- II. Throughput: The throughput of the system can be measured by calculating the number of requests processed per unit of time. This metric can be calculated as the ratio of the total number of requests processed to the total time in a given period (e.g., requests per second).
- III. Latency: The system's latency can be measured by calculating the time it takes to process a request. This metric is calculated as the average time it takes to complete a request from the time it is submitted (end-to-end delay).

- IV. Scalability: The scalability of the system can be determined by calculating the maximum number of concurrent requests that the system can handle without degrading performance. This metric can be calculated by increasing the number of requests gradually and measuring the system's response time (number of nodes, packets sent to BS, packet delivery ratio, number of destinations).

In conclusion, by measuring the performance of SPIN, GAF, and SPIN-G using these QoS attributes and corresponding metrics, we can identify areas for improvement and optimize the systems to provide better QoS to wireless sensor networks.

3.0.5 To compare and evaluate the performance of the SPIN-G against the SPIN and GAF protocols.

This research defines all of the specific performance matrices and conducts experiments or simulations to measure these matrices to compare and evaluate the performance of the SPIN-G protocol against the SPIN and GAF protocols.

Here are the potential steps used by this research:

- I. Define performance metrics: Before comparing protocol performance, you must define specific metrics to evaluate their effectiveness. In this study, for example, we consider metrics such as data delivery rate, end-to-end delay, throughput, packet loss, and so on.
- II. Design experiments or simulations: design experiments or simulations are based on the defined metrics to compare the performance of the protocols. To evaluate the performance of the protocols under various scenarios, you could consider different network topologies, traffic patterns, and network sizes.
- III. Implement the protocols: Implement the SPIN-G, SPIN, and GAF protocols in a MATLAB simulation environment.

- IV. Execute experiments: Execute the experiments or simulations designed in step II using the protocols that were implemented. Record and compare the performance metrics for each protocol.
- V. Analyze results: Examine the results of the experiments to determine the strengths and weaknesses of each protocol. Draw conclusions about which protocol outperforms the others under different conditions.
- VI. Conclusions: Based on the results and analysis, conclude the SPIN-G protocol's performance in comparison to the SPIN and GAF protocols. When concluding, consider scalability, throughput, bit error rate, data delivery rate, and so on.

Overall, comparing and evaluating the performance of different protocols requires a well-defined methodology and careful experimentation or simulation to ensure fair comparisons. Following that, the results are displayed in graphs, and the findings should demonstrate that the proposed protocol outperforms the existing protocols.

3.0.6 Implementation methods

In WSN, different models can be used to implement a specific method. The development of the emerging SPIN-G while incorporating the Dijkstra shortest path algorithm is the focus of this research implementation. This research carefully highlighted the parameters used in ensuring QoS by programming, the SPIN protocol, and the GAF location algorithm, and merging them with the Dijkstra shortest path algorithm using MATLAB (R2019b).

The non-clustered head architecture was used to create a clustered-based topology for SPIN in the first stage of implementation. Using random deployment for fifty (50) sensor nodes in a range of two hundred fifty meters (250m) and combining the SPIN clustered topology with the location-based algorithm GAF. The modified Dijkstra algorithm is fundamentally based on Dijkstra SPA to ensure the smooth transmission and delivery of data to the destination in the SPIN-G protocol.

The MATLAB simulator with the most vibrant graphical user interface (GUI) and advanced network parameters with custom-made functions was carefully chosen for the experiment. While carrying out the implementation task, we vigorously conducted the simulation in accordance with these phases:

3.0.0.1 Phase I: Sensor Nodes Transmission

To implement sensor nodes transmission it is dependent on the signal strength and distance. As a result, any event occurring within the sensing range can be detected by the sensor node at the location. In Figure 11, the transmission range was set at 250 meters, allowing 50 sensor nodes placed at random locations to communicate at a distance of 1000 meters by 1000 meters. Data is sensed and transmitted within a predefined range from the receiving node to the destination node or base station via the shortest path. The formula used to calculate the distance is given below:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \text{eq. 3.1}$$

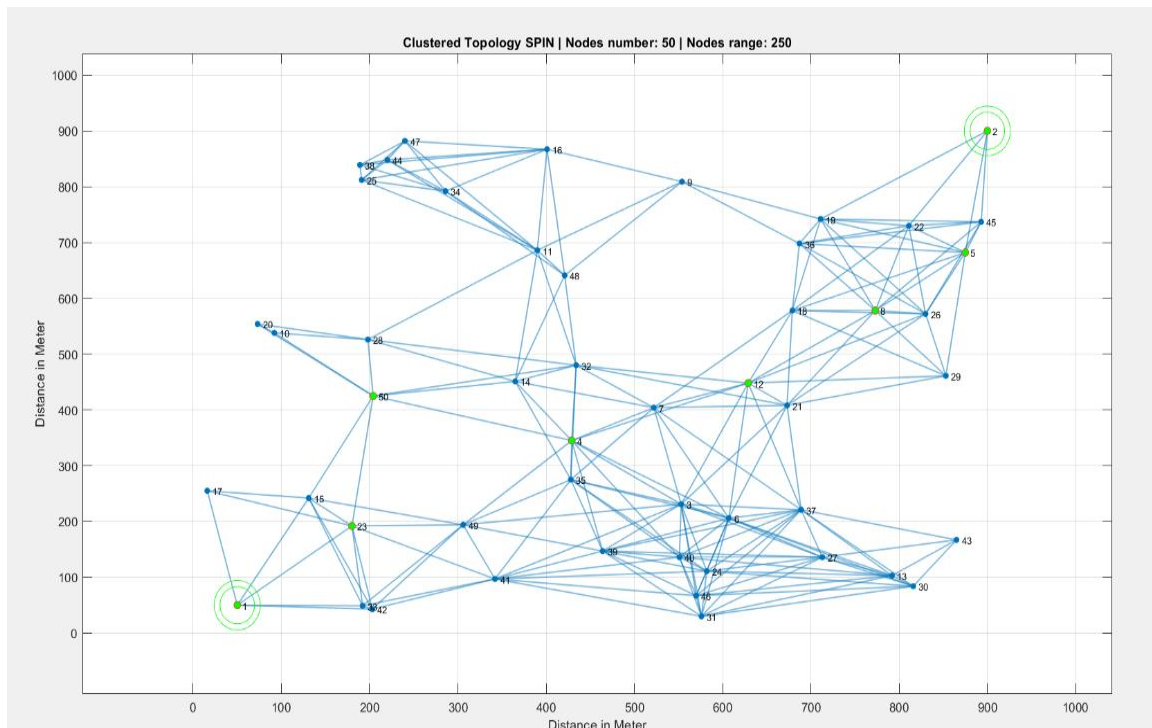


Figure 11: The implementation of Clustered-based topology for SPIN with Dijkstra Algorithm.

The steps taken by this study to initialize all 50 sensor nodes for the experiment in the SPIN clustered topology are as follows:

1. Deployed sensor nodes forming a non-clustered head architecture for SPIN: In this step, the sensor nodes were deployed in the network area. The non-clustered head architecture implies that all sensor nodes operate independently without forming any hierarchical clusters. Each sensor node functions as an individual entity in the network.
2. Set the maximum distance between the sensors to 250 meters: The study defined a maximum distance of 250 meters between the sensor nodes. This distance constraint ensures that the nodes are within communication range of each other. Nodes beyond this distance may not be able to establish direct communication links.
3. Formed clustered grid connectivity between all sensors: After deploying the sensor nodes, the study formed a clustered grid connectivity among them. The clustered grid topology organizes the sensor nodes into clusters based on their spatial proximity. This arrangement helps in efficient data dissemination and aggregation within each cluster.
4. Testing the modified Dijkstra algorithm in SPIN: Once the sensor nodes are initialized and organized into clusters, the experiment involves testing a modified version of the Dijkstra algorithm within the SPIN framework. The modified algorithm likely aims to optimize routing or other aspects of data transmission within the clustered network.

The exact details of how the clusters were formed and how the connectivity between the nodes was established might not be specified in the provided information. However, the goal of forming the clustered grid connectivity is to enable better communication and data management within the network, which can enhance the overall performance of the SPIN protocol. These steps suggest that the study aimed to investigate the performance of the SPIN protocol in a clustered topology with a specific maximum communication range between the nodes. This setup allows for the evaluation of SPIN's efficiency and effectiveness in a realistic network scenario. Overall, these steps outline the process of initializing the sensor nodes, defining communication parameters, establishing clustered grid connectivity, and subsequently testing a modified Dijkstra algorithm in the SPIN clustered topology.

3.0.0.2 Phase II: Data Delivery Mechanism using SPA

The acquisition and delivery of data is a critical component of this research's implementation for ensuring the data delivery mechanism addresses the data delivery problem. The sensor node is used by the data delivery mechanism to select the route to send data based on the locality of the destination and the locations of sensor nodes in the grid. The GAF algorithms for location-based use information about the location of data to help facilitate a smooth and reliable route for transmission.

To be more specific, figure 12 displays nodes that are close to each other or in the same grid and can share position data to ensure that location information is sent to all intended nodes in the network before a packet is forwarded. The source coordinates should be included in the packet to determine the destination. Geographical algorithms can obtain location information from a specific system, such as GPS, or virtual coordinates. Geo-casting is a method of delivering data to sensor nodes in a network grid using geographical protocols (van Hoesel et al., 2013).

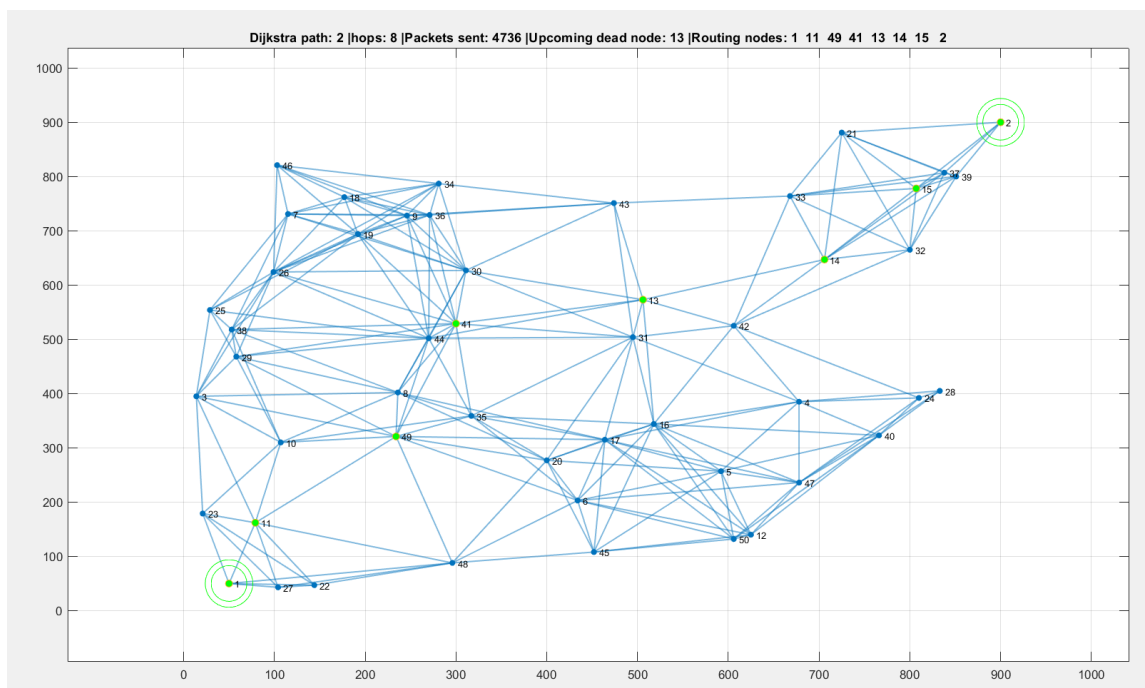


Figure 12: This diagram shows the implementation of modified Dijkstra in SPIN-GAF with 50 nodes.

3.0.0.3 Phase III: Implementing the parameters

The SPIN-G parameters refer to the scope of events or series of events that a specific condition is limited to addressing. The SPIN-G parameters were identified and implemented to improve data delivery quality of service (QoS) and overcome the SPIN setback. This study chose the parameters that will be used to compare the QoS in SPIN-G to the SPIN and GAF protocols. Packet loss, number of sensor nodes, end-to-end delay (latency), number of destinations, the packet sent to BS, signal noise ratio, bit error rate, throughput, packet delivery ratio, simulation time, and the number of sensors reduced are the parameters implemented. Using key metrics and appropriate MATLAB functions to create individual plots that can measure performance output in a graph.

3.1 Flowchart for the implemented approach

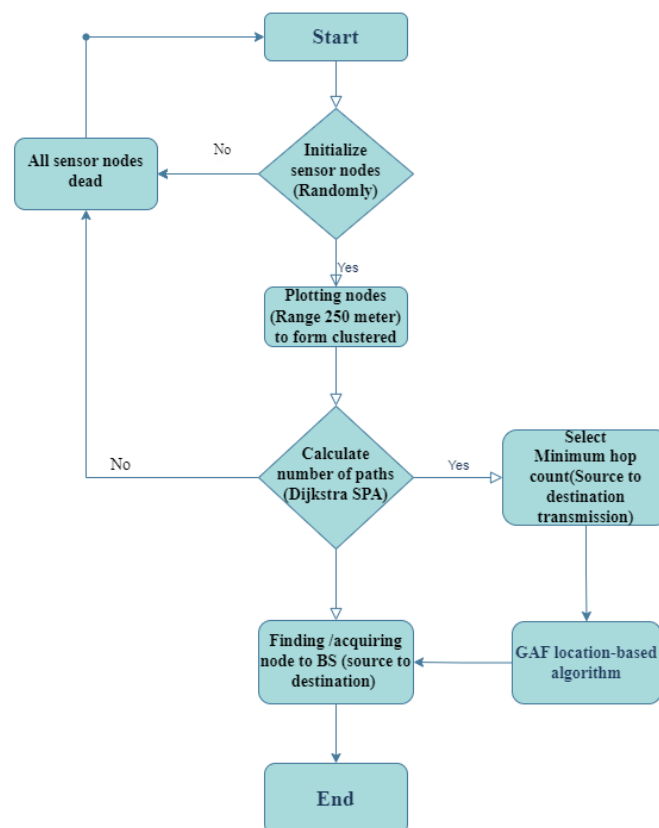


Figure 13: The flowchart diagram of the implementation of the proposed SPIN-G protocol.

3.2 Desirable approach

Wireless sensor networks (WSNs) indeed play a crucial role in various applications and ensuring Quality of Service (QoS) is essential to meet the specific requirements of these applications. This chapter focuses on investigating the QoS in WSNs and presents the results of the study. The first step in ensuring QoS in WSNs is to define the specific QoS requirements and objectives for the network. This involves determining the performance metrics and parameters that are critical for the application and designing the network accordingly. By deploying the sensor nodes accordingly in a desirable location, the network can benefit from better connectivity and reduced signal interference. This can lead to improved data transmission, lower packet loss, and enhanced overall network performance (E. Mbowe & S. Oreku, 2014). QoS-based routing is a common approach used in WSNs to optimize the utilization of network resources and meet specific QoS requirements. This routing strategy considers parameters such as network resources, delay time, and signal quality when making routing decisions. By selecting paths that meet the desired QoS criteria, the network can achieve better performance and ensure that the transmitted data satisfies the specified quality requirements (Kochhar et al., 2018).

Here are the key steps used by this study in ensuring QoS in WSNs:

1. **Requirement Analysis:** Understand the application requirements and identify the QoS parameters that are crucial for the application's success. This may include metrics such as latency, reliability, throughput, energy efficiency, and network lifetime.
2. **QoS Parameter Definition:** Clearly define the desired values or ranges for the identified QoS parameters. For example, specify the maximum acceptable latency, minimum required reliability, or desired throughput levels.
3. **Network Design:** Based on the QoS requirements, design the network architecture, including the placement of sensor nodes, routing protocols, and data aggregation techniques. Consider factors such as coverage, connectivity, energy efficiency, and load balancing.

4. **Resource Allocation:** Allocate network resources such as bandwidth, power, and memory to different tasks and nodes in a way that optimizes the QoS parameters. This may involve techniques like channel allocation, power control, and scheduling algorithms.
5. **QoS-Aware Routing:** Employ QoS-aware routing protocols that consider the QoS requirements and optimize the routing paths accordingly. These protocols should take into account factors like energy efficiency, traffic load balancing, and reliability.
6. **Traffic Management:** Implement traffic management mechanisms to prioritize critical data and allocate network resources accordingly. This may involve techniques such as traffic shaping, admission control, and Quality of Service (QoS) differentiation mechanisms.
7. **Monitoring and Feedback:** Continuously monitor the network performance and collect feedback on the QoS parameters. Use this information to analyze and optimize the system further, making adjustments as needed.

It's important to note that ensuring QoS in WSNs is a complex task that requires careful consideration of the application requirements, network design, resource allocation, and system optimization. Additionally, different applications may have different QoS requirements, so the specific steps and techniques may vary based on the application system requirement specifications and the desired performance objectives.

Building on the preceding of the previous chapter that delineated diverse methods employed for experimentation on the proposed SPIN-G protocol and the evaluation of SPIN-G in conjunction with previous research on the original SPIN, the suggested simulation was executed simultaneously with the existing data delivery predicament in SPIN. This research proposed the amalgamation of the GAF location-based algorithm with SPIN, alongside the implementation of a modified version of Dijkstra's shortest path algorithm (SPA), to guarantee a seamless transition and efficient data delivery, culminating in the development of the SPIN-G protocol.

3.3 Performance Metrics

A way of proving that the QoS is achieved through the Modified Dijkstra shortest path algorithm, and data delivery mechanism from GAF, this research conducted the simulation for all experiments to prove the significant improvements in the proposed SPIN-G protocol. A tool for simulation called MATLAB was used to achieve the specific objectives of this research. In other to evaluate the performance of the proposed protocol (SPIN-G) this research chose to consider the following parameters to evaluate the QoS from several other experiments conducted.

The performance evaluation parameters (metrics) used by this research to evaluate the protocols (SPIN, GAF, and SPIN-G) of interest to the study are:

1. Latency: in WSNs refers to the time delay between the generation of a data packet at a source node and its successful delivery to the destination node. The calculation of latency in WSNs depends on several factors, including network architecture, communication protocols, transmission distances, and processing delays. One common approach to estimating the latency in WSNs is by considering the transmission delay, propagation delay, and processing delay. The formula for calculating the latency can be expressed as:

$$\text{Latency} = \text{Transmission Delay} + \text{Propagation Delay} + \text{Processing Delay} \quad \text{eq. 3.2}$$

- I. Transmission Delay: It is the time taken to transmit a packet from the source node to the destination node. It can be calculated using the following formula:

$$\text{Transmission Delay} = \text{Packet Size} / \text{Data Rate} \quad \text{eq. 3.3}$$

Here, the packet size is the size of the data packet in bits or bytes, and the data rate is the transmission rate in bits per second (bps) or bytes per second (Bps).

- II. Propagation Delay: It is the time taken for a signal to travel from the source node to the destination node. It can be calculated using the following formula:

$$\text{Propagation Delay} = \text{Distance} / \text{Propagation Speed} \quad \text{eq. 3.4}$$

Here, the distance is the physical distance between the source and destination nodes, and the propagation speed is the speed at which signals propagate in the medium.

- III. Processing Delay: It is the time taken by the nodes to process the received packets, including the time for data processing, buffering, and routing. The processing delay can vary depending on the node's capabilities, network load, and the complexity of processing algorithms.

The specific values for packet size, data rate, distance, propagation speed, and processing delay would need to be determined based on the characteristics of the particular WSN deployment.

2. Throughput (TP): in WSNs refers to the rate at which data can be successfully transmitted from source nodes to destination nodes within the network. It represents the amount of data that can be transmitted per unit of time. The unit of measurement in terms of data packets is in kilobits per second (kbps). The calculation of throughput in WSNs depends on various factors, including network topology, communication protocols, channel conditions, and packet loss.

The formula for calculating throughput in WSNs is as follows:

$$\text{Throughput} = (\text{Total data received at the destination}) / (\text{Total time taken for transmission}) \quad \text{eq. 3.5}$$

To calculate the total data received at the destination, you need to consider the packet size and the number of successfully received packets. Let's denote the packet size as P_s and the number of successfully received packets as N .

$$\text{Total data received at the destination} = P_s * N \quad \text{eq. 3.6}$$

To calculate the total time taken for transmission, you need to consider the time taken to transmit all the packets successfully from the source to the destination. Let's denote the total time taken for transmission as T_t .

Total time taken for transmission = T_t

Substituting the values in the formula, we get:

$$\text{Throughput} = (P_s * N) / T_t \quad \text{eq. 3.7}$$

The packet size (P_s) is usually measured in bits or bytes, the number of successfully received packets (N) is a unitless count, and the total time taken for transmission (T_t) is measured in seconds. It's important to note that the actual throughput in a WSN may be lower than the theoretical throughput due to various factors such as network congestion, interference, collisions, and routing overhead. Therefore, the calculated throughput may serve as an estimate rather than an absolute value.

3. Packet loss (PL): in WSNs refers to the percentage of packets that are not successfully received at the destination node (Said, 2015). It represents the loss of data during transmission due to various factors such as network congestion, interference, collisions, poor signal, and transmission errors. Lower packet loss is desirable as it ensures reliable data delivery. The calculation of packet loss in WSNs can be done using the following formula:

$$\text{Packet Loss} = (\text{Number of Lost Packets} / \text{Total Number of Sent Packets}) * 100 \quad \text{eq. 3.8}$$

To calculate the packet loss, you need to know the number of lost packets and the total number of sent packets.

- I. Number of Lost Packets: It refers to the count of packets that were not successfully received at the destination node. This can be obtained by monitoring the network and identifying the packets that were lost during transmission.

- II. Total Number of Sent Packets: It refers to the count of all the packets sent from the source node to the destination node. This can be obtained by tracking the packets generated by the source node and ensuring that they are properly accounted for. Once you have the values for the number of lost packets and the total number of sent packets, you can plug them into the formula to calculate the packet loss percentage. For example, if you have sent 100 packets and 5 of them were lost during transmission, the calculation would be:

$$\text{Packet Loss} = (5 / 100) * 100 = 5\% \quad \text{eq. 3.9}$$

This means that the packet loss in the WSNs is 5%.

It's worth noting that packet loss can vary over time and under different network conditions. Monitoring and analyzing packet loss is crucial for assessing the performance and reliability of WSNs and may help identify areas for improvement.

4. Packet delivery ratio (PDR): in WSNs refers to the ratio of successfully received packets at the destination node to the total number of packets sent from the source node. It represents the effectiveness of packet delivery in the network. A higher packet delivery ratio implies better QoS in wireless sensor networks. By measuring this metric for SPIN, GAF, and SPIN-G, you can assess the improvement in packet delivery for SPIN-G compared to the original SPIN protocol. The reported 99.1% successful packet delivery rate for SPIN-G over SPIN suggests a significant enhancement in data delivery efficiency.

The calculation of the packet delivery ratio in WSNs can be done using the following formula:

$$\text{Packet Delivery Ratio} = (\text{Number of Received Packets} / \text{Number of Sent Packets}) * 100 \quad \text{eq. 3.10}$$

To calculate the packet delivery ratio, you need to know the number of received packets and the number of sent packets.

- I. Number of Received Packets: It refers to the count of packets that were successfully received at the destination node. This can be obtained by monitoring the network and identifying the packets that reached their intended destination.
- II. Number of Sent Packets: It refers to the count of all the packets sent from the source node to the destination node. This can be obtained by tracking the packets generated by the source node and ensuring that they are properly accounted for. Once you have the values for the number of received packets and the number of sent packets, you can plug them into the formula to calculate the packet delivery ratio.

For example, if you have sent 100 packets and 95 of them were successfully received at the destination node, the calculation would be:

$$\text{Packet Delivery Ratio} = (95 / 100) * 100 = 95\% \quad \text{eq. 3.11}$$

This means that the packet delivery ratio in the WSN is 95%.

Calculating the packet delivery ratio helps evaluate the performance and reliability of the QoS in SPIN-G, indicating the effectiveness of the communication and the presence of any issues such as packet loss or network congestion.

5. Signal-to-Noise Ratio (SNR): SNR in wireless sensor networks measures the strength of the desired signal relative to the background noise present in the network which is measured as a single numerical value in decibels (dB). It is a ratio that quantifies the quality of the signal being transmitted. SNR is an important factor in determining the quality of wireless communication. You can measure the SNR for the protocols and compare them to evaluate their performance in terms of signal quality.

The calculation of SNR in WSNs can be done using the following formula:

$$\text{SNR} = 10 * \log_{10}(\text{Ps} / \text{Pn}) \quad \text{eq. 3.12}$$

Where:

- SNR is the signal-to-noise ratio in decibels (dB).
- Ps is the power of the signal.
- Pn is the power of the noise.

To calculate the SNR, you need to know the power of the signal (Ps) and the power of the noise (Pn). These power values can be obtained through measurements or simulations based on the specific WSNs scenario. Once you have the power values, you can substitute them into the formula and calculate the SNR. The result is typically expressed in decibels (dB), which provides a logarithmic scale for the SNR measurement. It's important to note that the SNR value indicates the quality of the signal and its ability to overcome the noise.

Higher SNR values indicate better signal quality, while lower SNR values imply a weaker signal that is more susceptible to noise interference. Keep in mind that SNR calculations can be more complex depending on the specific characteristics of the wireless channel, modulation schemes, and interference sources in the WSNs. Therefore, additional considerations may be required to account for factors such as fading, interference, and channel impairments in more complex scenarios.

6. The Bit Error Rate (BER): in wireless sensor networks is a measure of the quality of the received signal. It represents the ratio of the number of erroneous bits to the total number of transmitted bits. The BER is typically calculated using the following formula:

$$\text{BER} = (\text{Number of erroneous bits}) / (\text{Total number of transmitted bits}) \quad \text{eq. 3.13}$$

To calculate the BER, you need to know the number of erroneous bits and the total number of transmitted bits. The number of erroneous bits can be obtained by comparing the received bits with the original transmitted bits, while the total number of transmitted bits is simply the number of bits that were sent. Keep in mind that the BER calculation assumes that the transmitted bits are known and can be compared with the received bits. In practice, this requires a mechanism for error detection or error correction, such as the use of error detection codes or forward error correction techniques.

It's also important to note that the BER calculation does not take into account the specific characteristics of the wireless channel, such as fading, interference, or noise. These factors can significantly affect the error rate in wireless communication systems and may require additional models or simulations to accurately estimate the BER in WSNs.

3.4 Overall Summary of Methodology

In this chapter, the main aim and objectives of the study were applied to enhance the Quality of Service (QoS) in Wireless Sensor Networks (WSNs). This research utilized various methods to develop the SPIN-G protocol, which involved integrating the SPIN and GAF location-based algorithms. The implementation of this research was carried out using MATLAB, ensuring all necessary steps and procedures were followed to achieve successful and error-free simulation procedures.

One of the key accomplishments of this study was the successful implementation of a clustered-based topology for SPIN-G. This was achieved by randomly placing all nodes in the clustered grid, which contributed to the overall effectiveness of the protocol. Additionally, modified Dijkstra's algorithm was incorporated into SPIN-G to calculate the shortest routes within the network.

To evaluate the QoS of the SPIN-G protocol, this research designed performance metrics. These metrics served as a means to test the effectiveness and efficiency of the protocol, providing quantitative measurements to assess its QoS capabilities. By comparing the results of these performance metrics with those of the existing SPIN and GAF protocols, it was demonstrated that SPIN-G exhibited a significant improvement in QoS.

Overall, the methods employed in this research were deemed necessary and instrumental in achieving the desired improvement in QoS for the SPIN-G protocol. The integration of SPIN and GAF algorithms, the implementation of a clustered-based topology, the modification of Dijkstra's algorithm, and the design of performance metrics collectively contributed to the successful development and evaluation of the SPIN-G protocol in enhancing QoS in WSNs.

Chapter 4

4.0 Simulation Results and Discussion

4.1 Experimental Results

The system used for running the experiments was a computer with Windows 11 Pro, an AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx processor embedded with 4 cores and 8 logical processors running at 2.30GHz, and 8GB of main memory. The experiment involved 50 sensor nodes randomly positioned at a distance of 1000 meters by 1000 meters range. The experiment likely involved running simulations or implementing the SPIN, GAF, SPIN-G, and Dijkstra algorithms in MATLAB R2019b to evaluate their performance in terms of various metrics such as signal noise ratio, data delivery ratio, latency, or any other relevant performance indicators.

The objective of the experiments is to assess whether the proposed SPIN-G protocol improved the Quality of Service (QoS) in the WSNs and compare those individual protocols (SPIN, GAF) to the proposed SPIN-G protocol. It's important to note that without access to the specific source code and experimental details, it is not possible to provide specific insights into the results or conclusions of the experiment. However, based on the information provided, it appears that the experiments aimed to assess the performance improvements of the SPIN-G protocol compared to the individual SPIN and GAF protocols in terms of QoS in the WSNs scenario.

Table 2: Outline in this table are the initialized and output values for SPIN-G with Modified Dijkstra.

Results	
Area	1000m x1000m
Number of Sensor Nodes	50
Packet sent	4648kb
Min initial energy for sensor nodes	80 milliwatts (Mw)
Max initial energy for sensor nodes	100 milliwatts (Mw)
Energy consumption per circle	0.35 kilojoules (kJ)
Number of hops	7
Modified Dijkstra's paths	2
Transmission range	250 meters(m)

In the non-clustered head architecture, the sensor nodes are deployed in a clustered grid topology. In Figure 14, sensor node one (1) is depicted as the source node, while sensor node two (2) is represented as the destination node or base station. This visual representation suggests that a packet has been successfully delivered from the source node to the destination node through the involvement of various sensor nodes in the transmission process.

The clustered grid topology offers several advantages in wireless sensor networks. Firstly, it enables efficient data aggregation and processing within each cluster. The cluster member can collect data from its member nodes, perform local processing, and transmit aggregated information to the base station or sink node, reducing the overall energy consumption of the network. The simulation shown in Figure 14 is one of the many iterations and for every iteration, there's a new path discovered which is done by the Modified Dijkstra algorithm to determine the shortest possible route to transmit data in this clustered topology implemented for SPIN-G.

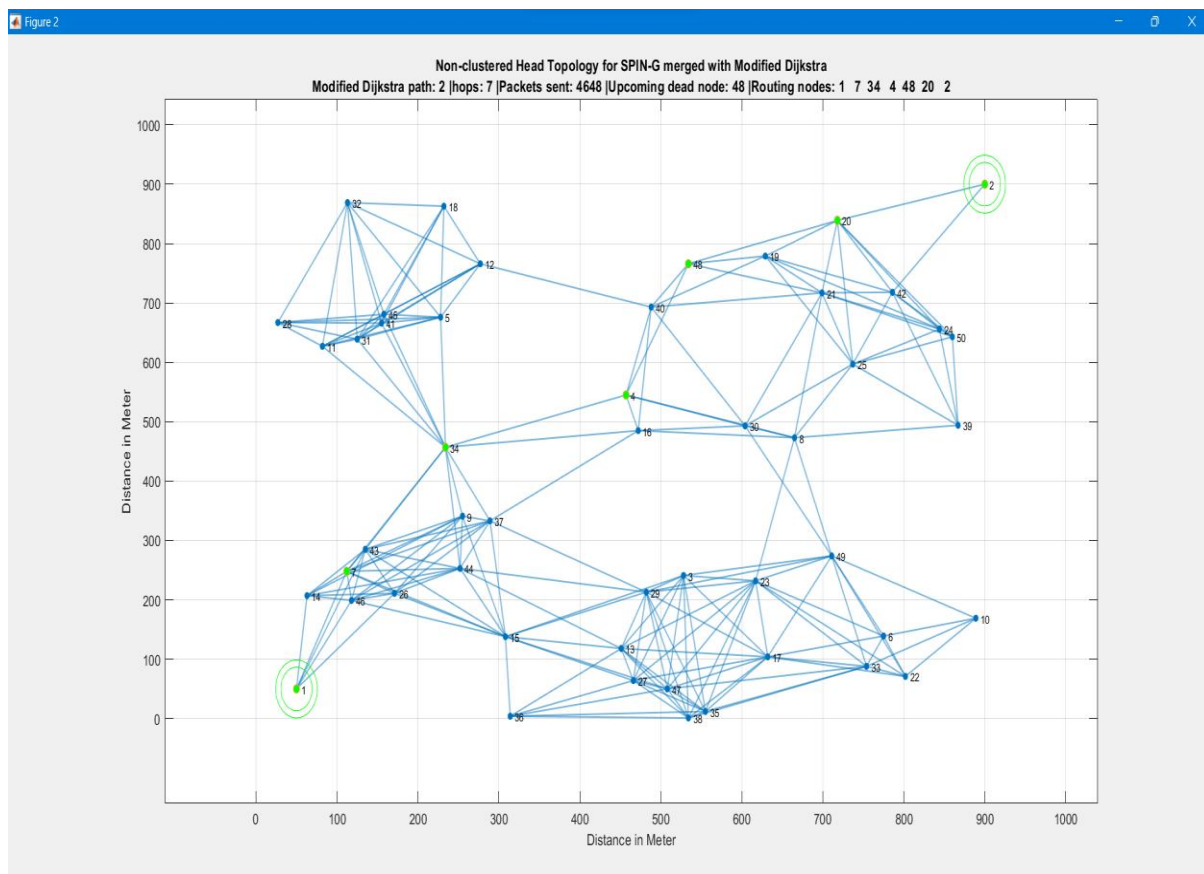


Figure 14: Modified Dijkstra implemented in SPIN-G showing the interlink of sensors and packets sent from source to destination (Iteration 1).

Table 3: Initialized and output values for Modified Dijkstra with SPIN-G

Results	
Area	1000m x1000m
Number of Sensor Nodes	50
Packet sent	4992kb
Min initial energy for sensor nodes	80 milliwatts (Mw)
Max initial energy for sensor nodes	100 milliwatts (Mw)
Energy consumption per circle	0.35 kilojoules (kJ)
Number of hops	8
Modified Dijkstra paths	3
Transmission range	250m
Dead nodes	Node 48

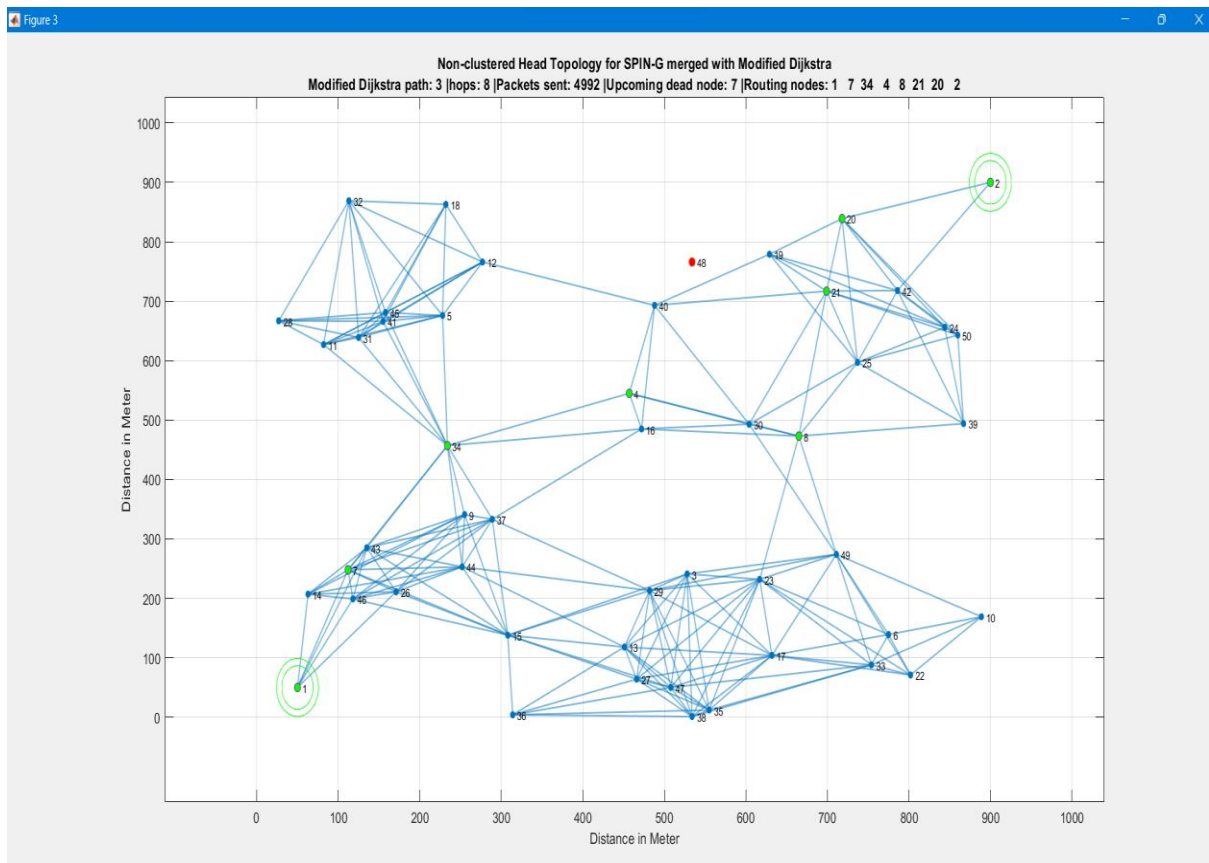


Figure 15: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 2).

Table 4: Outline in this table are the initialized and output values for Modified Dijkstra in SPIN-G.

Results	
Area	1000m x 1000m
Number of Sensor Nodes	50
Packet sent	5920kb
Min initial energy for sensor nodes	80 milliwatts (Mw)
Max initial energy for sensor nodes	100 milliwatts (Mw)
Energy consumption per circle	0.35 kilojoules (kJ)
Number of hops	8
Modified Dijkstra paths	4
Transmission range	250m
Dead nodes	Node 7,48

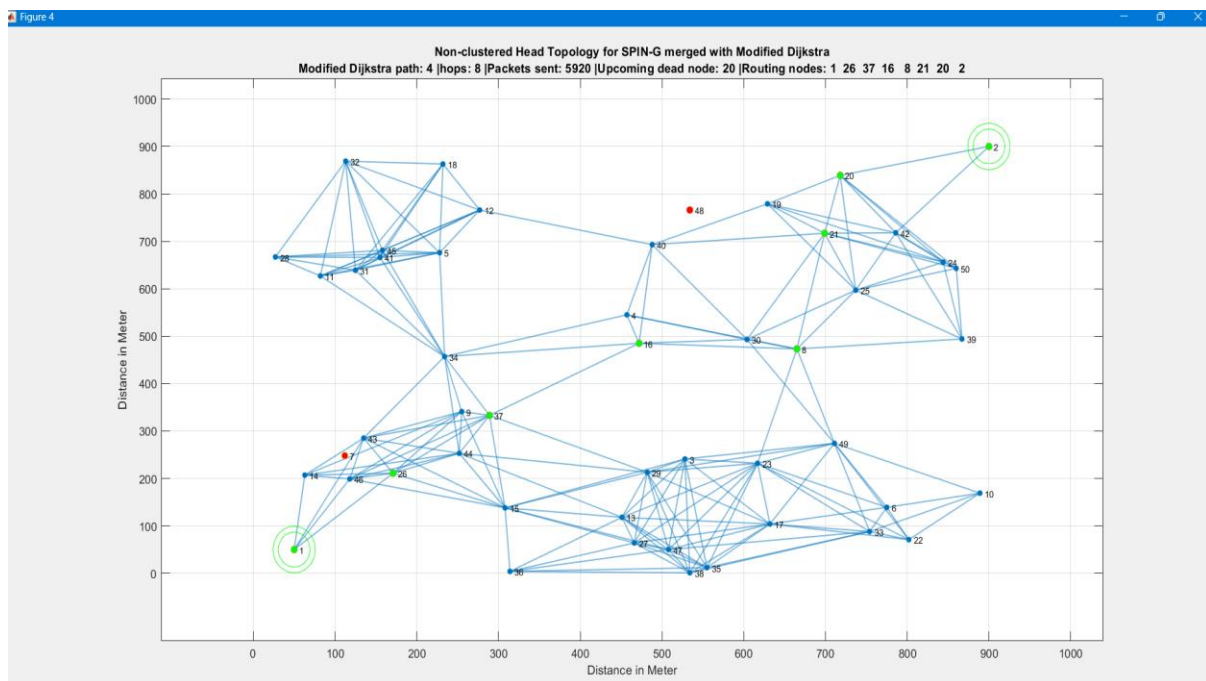


Figure 16: Modified Dijkstra implemented on SPIN-G showing the dead node and the hops (Iteration 3)

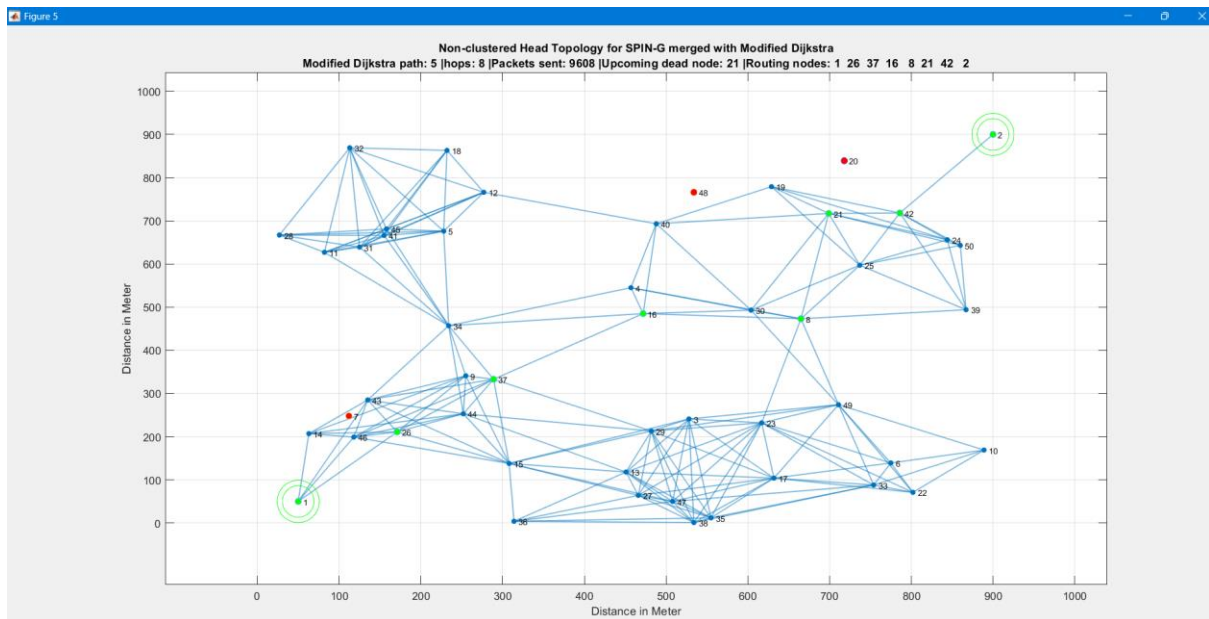


Figure 17: Modified Dijkstra implemented in SPIN-G showing the dead node and the hop count (Iteration 4)

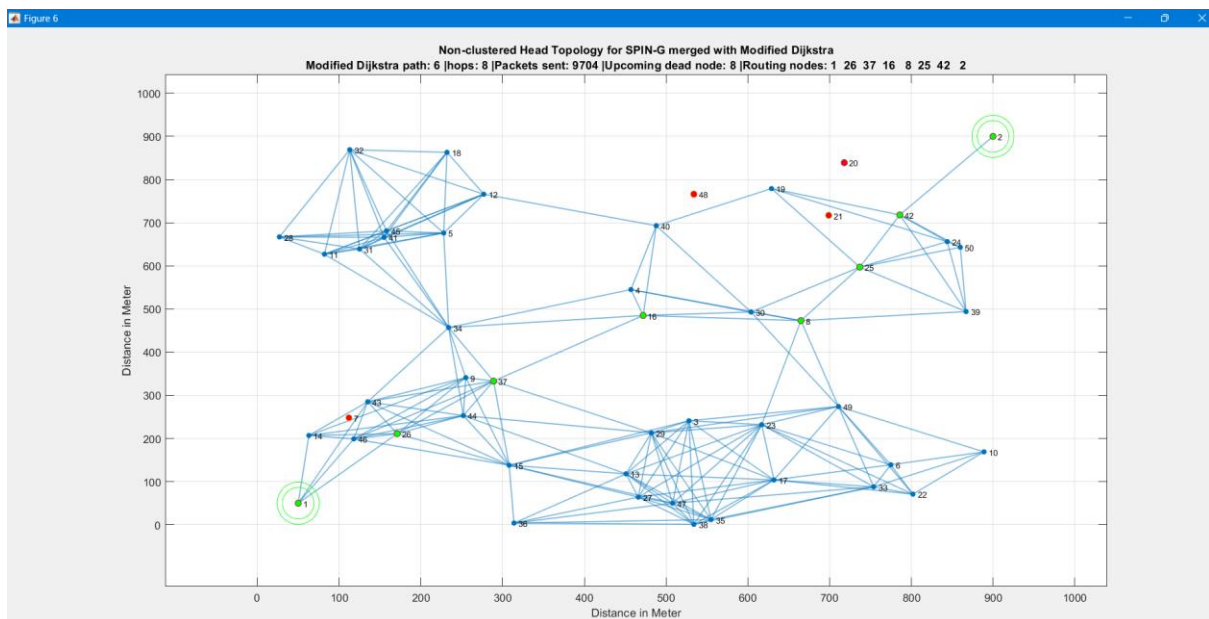


Figure 18: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 5)

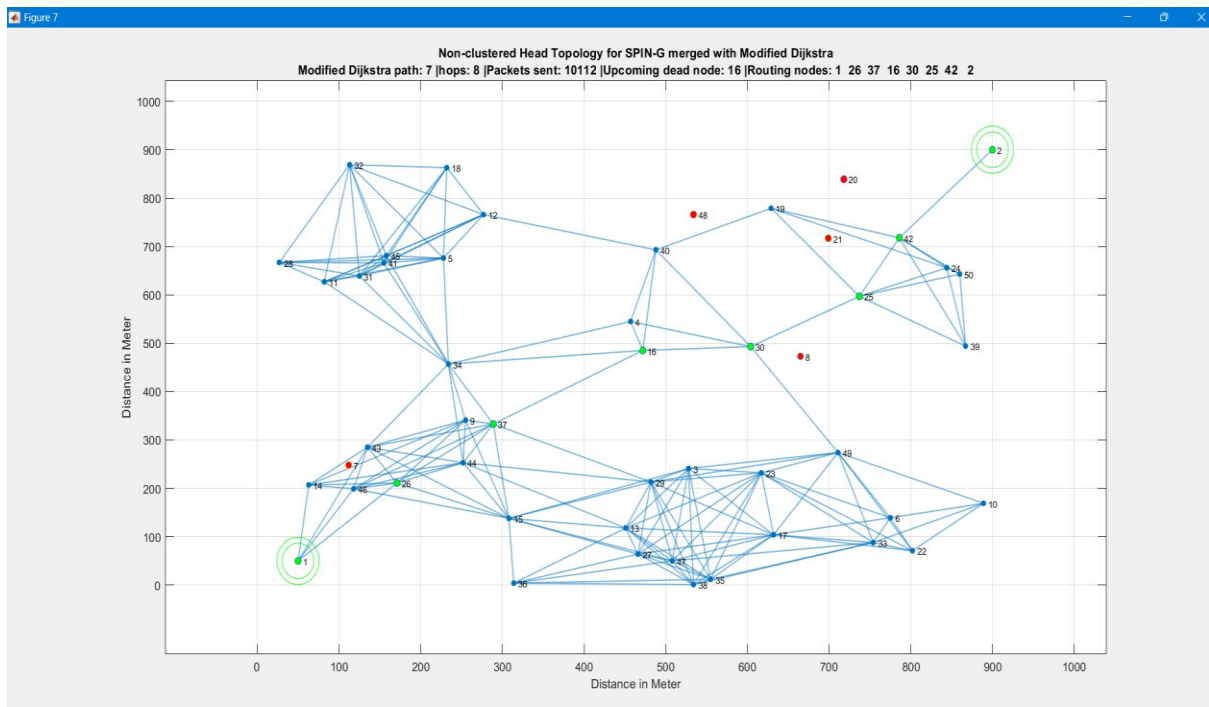


Figure 19: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 6)

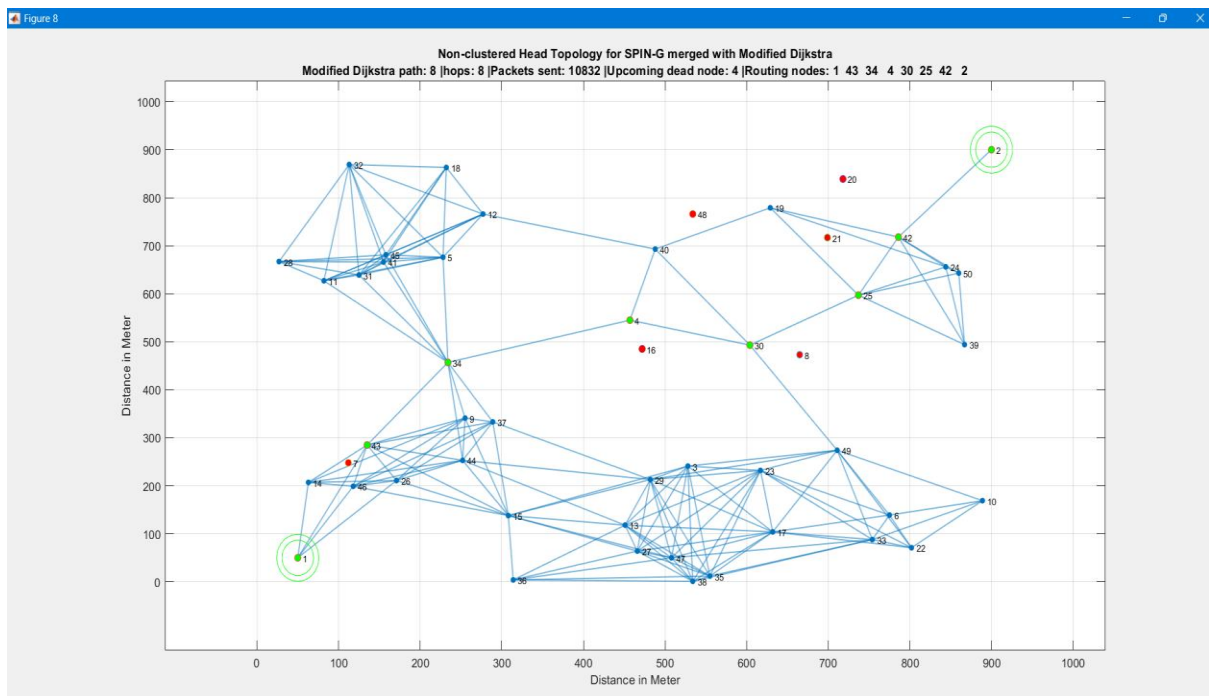


Figure 20: Modified Dijkstra implemented to SPIN-G showing the dead node and the hop count (Iteration 7)

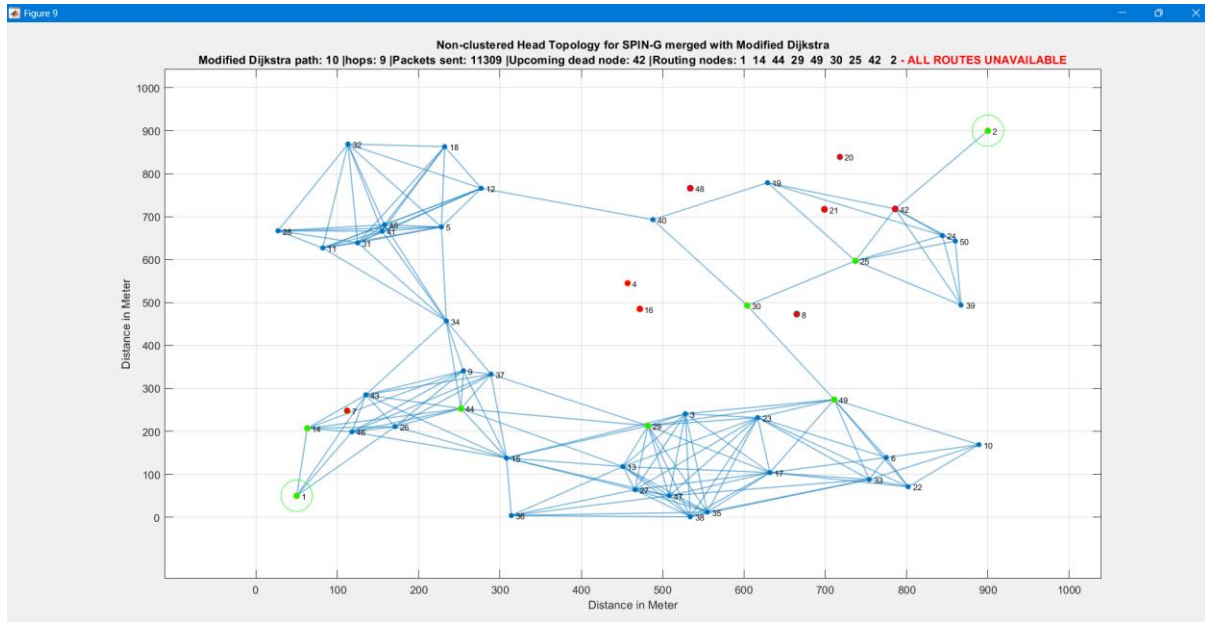


Figure 21: Modified Dijkstra implemented to SPIN-G capture no connection between source to the destination due to dead nodes and out-of-range nodes (Iteration 8).

In Figure 21 the simulation shows how possible network or connection failure may occur due to dead nodes in the network. The modified Dijkstra algorithm checked for all available paths and found out that there were none because of dead or inactive nodes in the transmission range. There's no possibility of the nearest node establishing contact because they are out of range.

4.2 Performance Evaluation of Protocols

In terms of performance evaluation, the SPIN-G protocol underwent a comprehensive comparative analysis with existing protocols, namely SPIN and GAF. The assessment encompassed multiple crucial metrics, including packet loss, number of sensor nodes, end-to-end delay, number of destinations, number of packets sent to the base station (BS), and Signal-to-Noise Ratio (SNR). By examining these factors, researchers aimed to ascertain the strengths and weaknesses of the SPIN-G protocol in relation to its counterparts.

First and foremost, packet loss, a vital parameter indicating the rate of data loss during transmission, was measured for the SPIN-G protocol alongside SPIN and GAF. This comparison allowed researchers to gauge the effectiveness of SPIN-G in maintaining data integrity and minimizing loss during communication. By analyzing the packet loss rates of the protocols, they could assess the robustness and reliability of SPIN-G.

Additionally, the number of sensor nodes, which refers to the number of network nodes responsible for sensing and transmitting data, was considered in the performance evaluation. This metric allowed researchers to understand the scalability and adaptability of the SPIN-G protocol in different network sizes. Comparing the number of sensor nodes required by SPIN-G with those of SPIN and GAF helped assess the efficiency and resource utilization of SPIN-G in diverse deployment scenarios.

The end-to-end delay, another significant metric, represents the time taken for a packet to travel from the source sensor node to its intended destination. By measuring the end-to-end delay for SPIN-G, SPIN, and GAF, researchers could determine the protocols' respective abilities to ensure timely and responsive data transmission. This evaluation shed light on the suitability of SPIN-G for applications requiring low-latency communication.

Furthermore, the number of destinations, indicating the total count of distinct recipient nodes in the network, was taken into account during the performance analysis. Comparing this metric across SPIN-G, SPIN, and GAF helped researchers understand the versatility and applicability of the protocols for scenarios involving multiple data recipients. This assessment provided insights into the efficiency and adaptability of SPIN-G when dealing with various destination nodes.

Moreover, the number of packets sent to the base station (BS) was examined as part of the performance evaluation. This metric reflects the volume of data forwarded to the central BS within the network. Comparing the packet count transmitted by SPIN-G, SPIN, and GAF enabled researchers to determine the protocols' respective data handling capacities and efficiency in aggregating sensor data at the BS. This analysis contributed to understanding the suitability of SPIN-G for applications with high data collection requirements.

Lastly, the signal-to-noise ratio (SNR), a measure of signal quality in the presence of background noise, was considered in the comparative evaluation. Assessing the SNR for SPIN-G, SPIN, and GAF allowed researchers to ascertain the protocols' abilities to maintain data integrity and reliability despite environmental noise. This examination aided in understanding the robustness and resilience of SPIN-G in challenging and noisy wireless communication environments.

By conducting a comprehensive performance evaluation encompassing packet loss, number of sensor nodes, end-to-end delay, number of destinations, number of packets sent to the BS, and Signal-to-Noise Ratio, this research was able to assess the efficacy and suitability of the SPIN-G protocol in comparison to its predecessors, SPIN, and GAF. This analysis provided valuable insights into the strengths and weaknesses of SPIN-G, enabling researchers and practitioners to make informed decisions regarding its adoption and implementation in wireless sensor networks.

4.3 Packet Loss

In a WSN, communication between devices is accomplished through the transmission of packets from one node to another. Sometimes the total number of packets sent is not delivered to the destination. The process of some packets failing to be delivered due to constraints such as power failure, network congestion, and poor signal strength may be the most likely cause of packet loss (Cirstea et al., 2012). The plot in Figure 22 depicts the assessment of SPIN, GAF, and SPIN-G performance with respect to the number of nodes, focusing on packet loss.

Table 5: Performance check for GAF, SPIN, and SPIN-G protocols by evaluating packet loss against the number of sensor nodes.

Protocols		Number of sensor nodes				
		5 Nodes	10 Nodes	15 Nodes	20 Nodes	25 Nodes
GAF	Packet loss (%)	7.067%	9.968%	11.8%	13.03%	13.97%
SPIN		8.481%	11.96%	14.16%	15.64%	16.77%
SPIN-G		5.654%	7.974%	9.44%	10.43%	11.18%

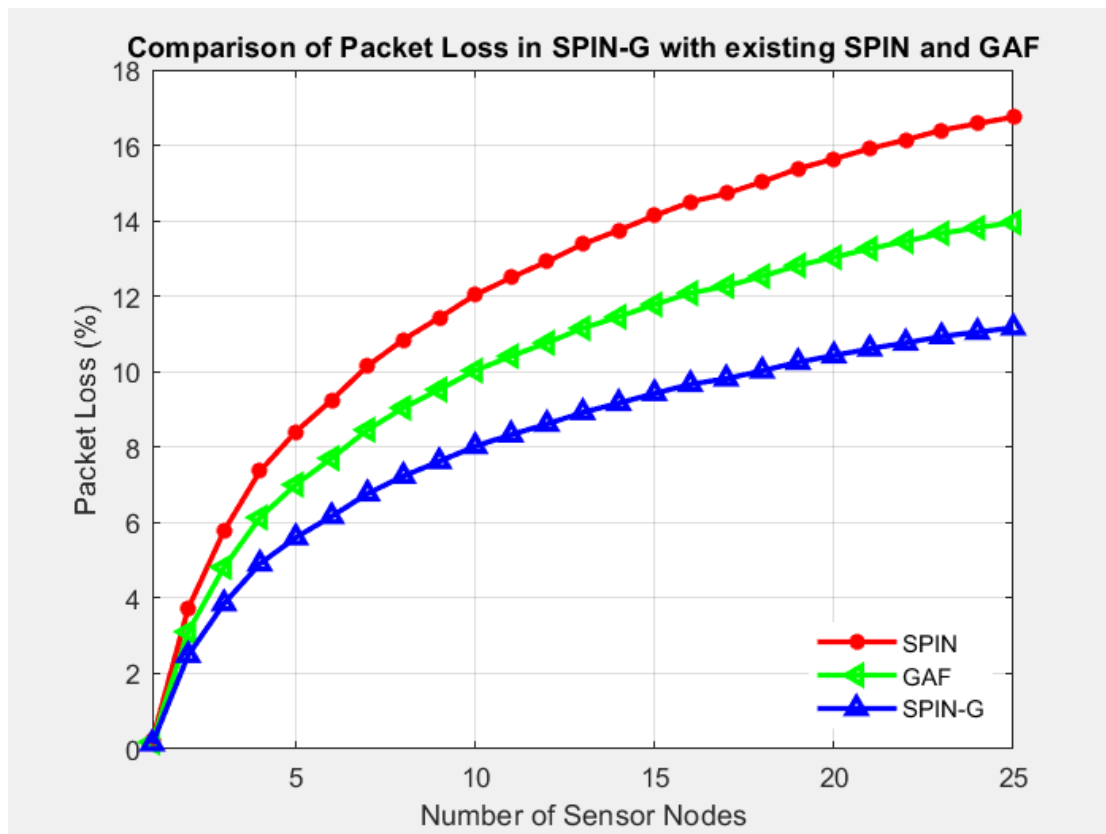


Figure 22:Evaluating the QoS in WSNs through packet loss vs the number of sensor nodes.

The plot shown in Figure 23 provides an evaluation of the performance of three protocols: SPIN, GAF, and SPIN-G. This evaluation specifically focuses on their performance in terms of packet loss, measured against the packet delivery rate. The plot allows us to observe and compare how these protocols handle packet loss under varying conditions of the packet delivery rate. By analyzing this plot, we gain insights into the effectiveness of the SPIN-G protocol and how it made improvements in packet delivery rate that better the QoS for WSNs.

Table 6: Performance check for GAF, SPIN, and SPIN-G protocols by evaluating packet loss against packet delivery rate.

Protocols		Packet Delivery Ratio				
	Packet loss (%)	20%	40%	60%	80%	100%
GAF		5.734	4.785	3.722	2.613	1.573
SPIN		6.449	5.618	4.652	3.579	2.471
SPIN-G		2.867	2.392	1.861	1.306	0.7865

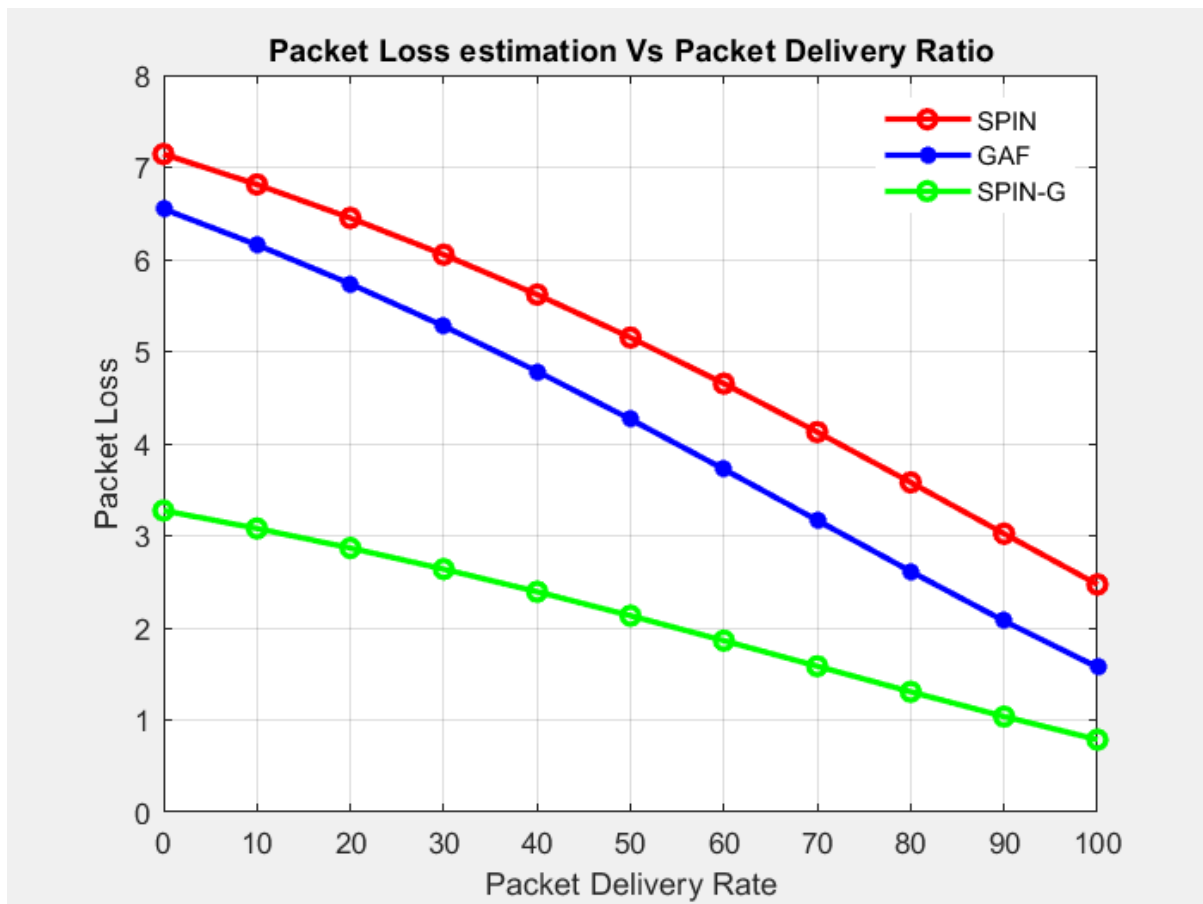


Figure 23: Using packet loss vs. packet delivery ratio to assess QoS in WSNs.

4.4 Latency

A delay in any network is critical for monitoring network performance. In Figure 24, the end-to-end delay is measured against the number of sensor nodes to evaluate the latency performance of the SPIN, GAF, and SPIN-G protocols. The table below shows SPIN-G performance as the lowest latency among the SPIN and GAF protocols from 5 to 25 nodes. This indicates that the SPIN-G protocol outperforms the SPIN and GAF protocols.

Table 7: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the end-to-end delay against the number of sensor nodes.

Protocols		Number of sensor nodes				
	End-to-end Delay	5 Nodes	10 Nodes	15 Nodes	20 Nodes	25 Nodes
GAF		0.02322ms	0.008209ms	0.002727ms	0.0008227ms	0.000278ms
SPIN		0.03483ms	0.01231ms	0.004091ms	0.001234ms	0.000417ms
SPIN-G		0.01742ms	0.006157ms	0.002046ms	0.000617ms	0.0002085ms

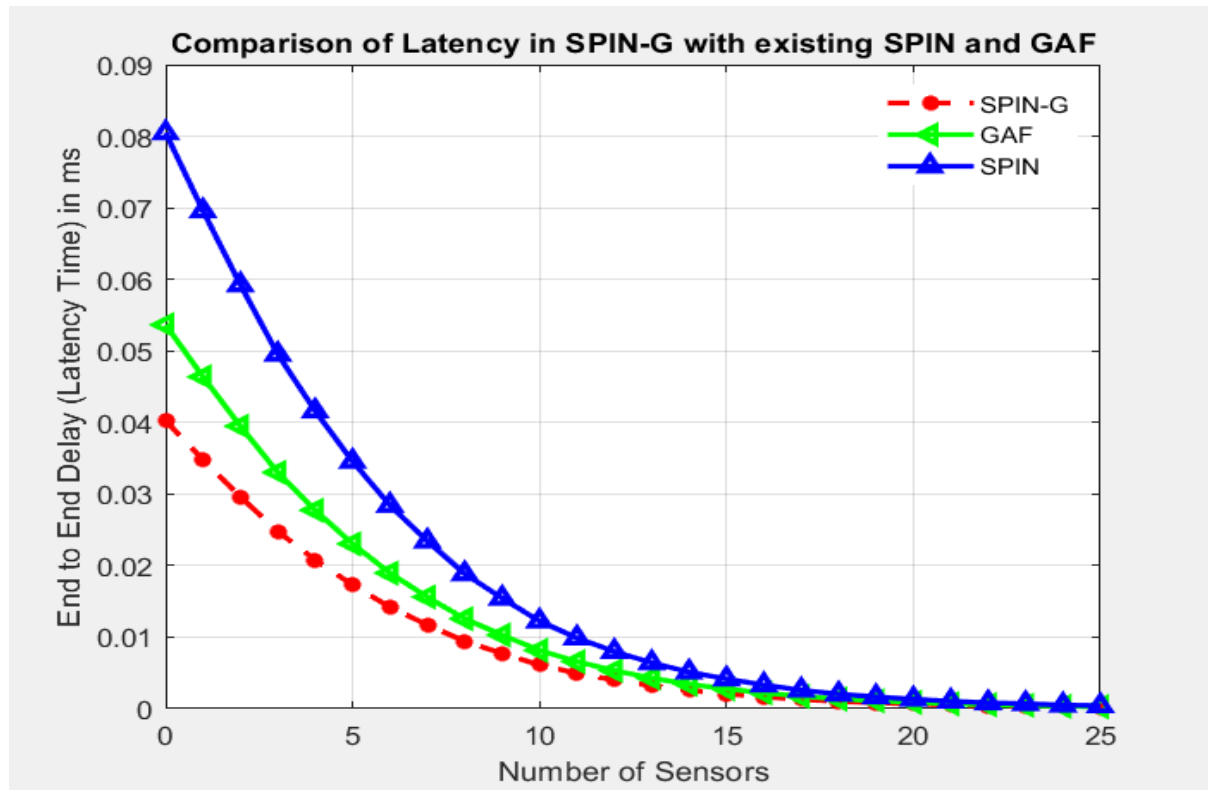


Figure 24: Evaluating the QoS in WSN through end-to-end delay vs the number of sensor nodes.

Monitoring network performance necessitates careful attention to any delays present within the network. Figure 25 presents a graphical representation where the end-to-end delay is measured in relation to the number of sensor destinations. This evaluation aims to assess the latency performance of three protocols: SPIN, GAF, and SPIN-G. The table accompanying the figure highlights the performance of the SPIN-G protocol, exhibiting the lowest latency among the SPIN and GAF protocols when the number of destinations increased from 1 to 7. This result signifies that the SPIN-G protocol surpasses the SPIN and GAF protocols in terms of latency performance.

Table 8: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the end-to-end delay against the number of destinations.

Protocols	End-to-end Delay	Number of Destinations						
		1	2	3	4	5	6	7
GAF		0.02806ms	0.01882ms	0.01144ms	0.00639ms	0.002994ms	0.001173ms	0.0004048ms
SPIN		0.05613ms	0.03763ms	0.02288ms	0.01278ms	0.005988ms	0.002346ms	0.0008096ms
SPIN-G		0.01403ms	0.009409ms	0.005719ms	0.003195ms	0.001497ms	0.0005865ms	0.0002024ms

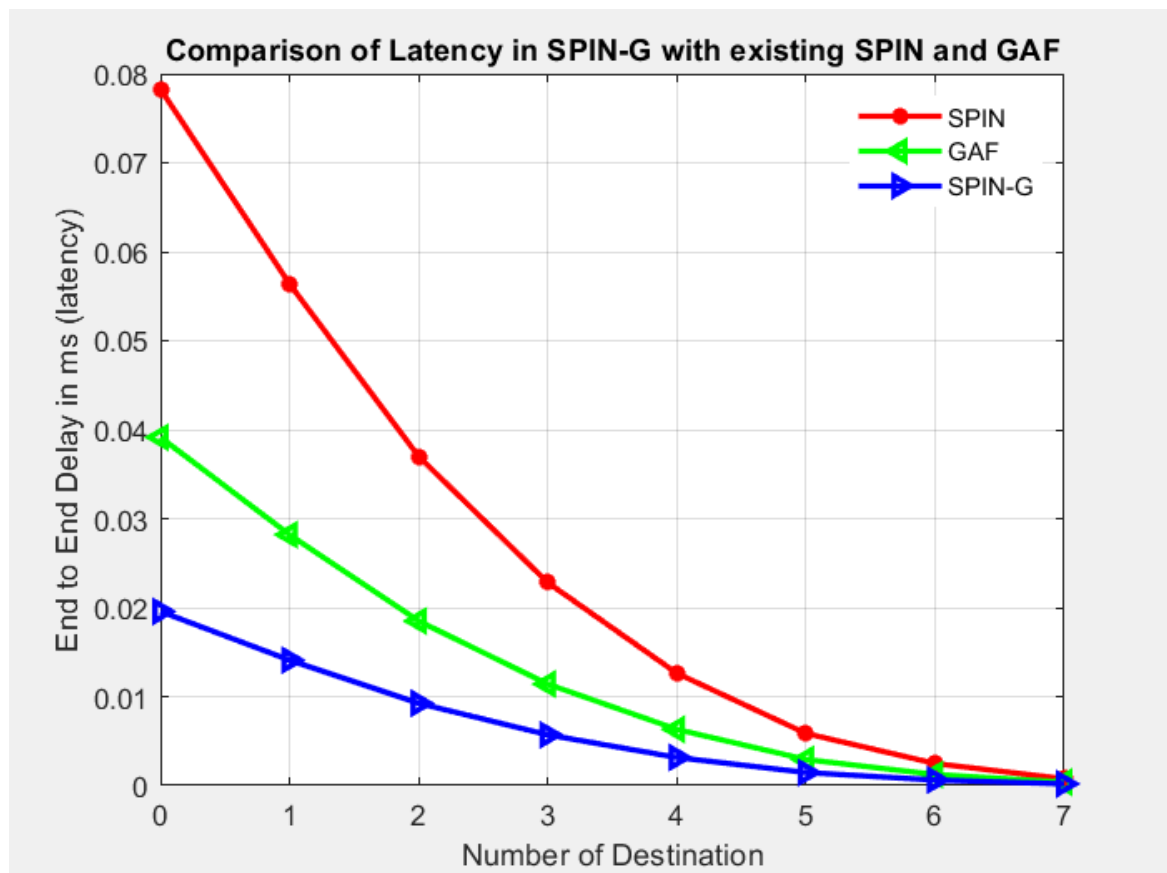


Figure 25:Evaluating the QoS in WSN through end-to-end delay vs the number of destinations.

4.5 Throughput

Throughput refers to the average duration taken for a packet to reach its destination successfully. Table 9 presents an evaluation of three protocols: SPIN, GAF, and SPIN-G, based on the number of packets transmitted to the base station in relation to the signal-to-noise ratio. In this simulation, the results obtained from the proposed SPIN-G protocol, as depicted in Figure 26, demonstrate a significantly higher number of successfully delivered packets compared to both SPIN and GAF. This indicates the superior performance of the SPIN-G protocol in terms of packet delivery.

Table 9: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the packets sent to the base station against the number of destinations.

Protocols	Packets sent to BS(%)	Signal Noise Ratio (dB)				
		5dB	10dB	15dB	20dB	25dB
GAF		43	57.6	66.13	72.52	77.86
SPIN		35.18	47.13	54.1	59.34	63.71
SPIN-G		54.72	73.31	84.16	92.3	99.1

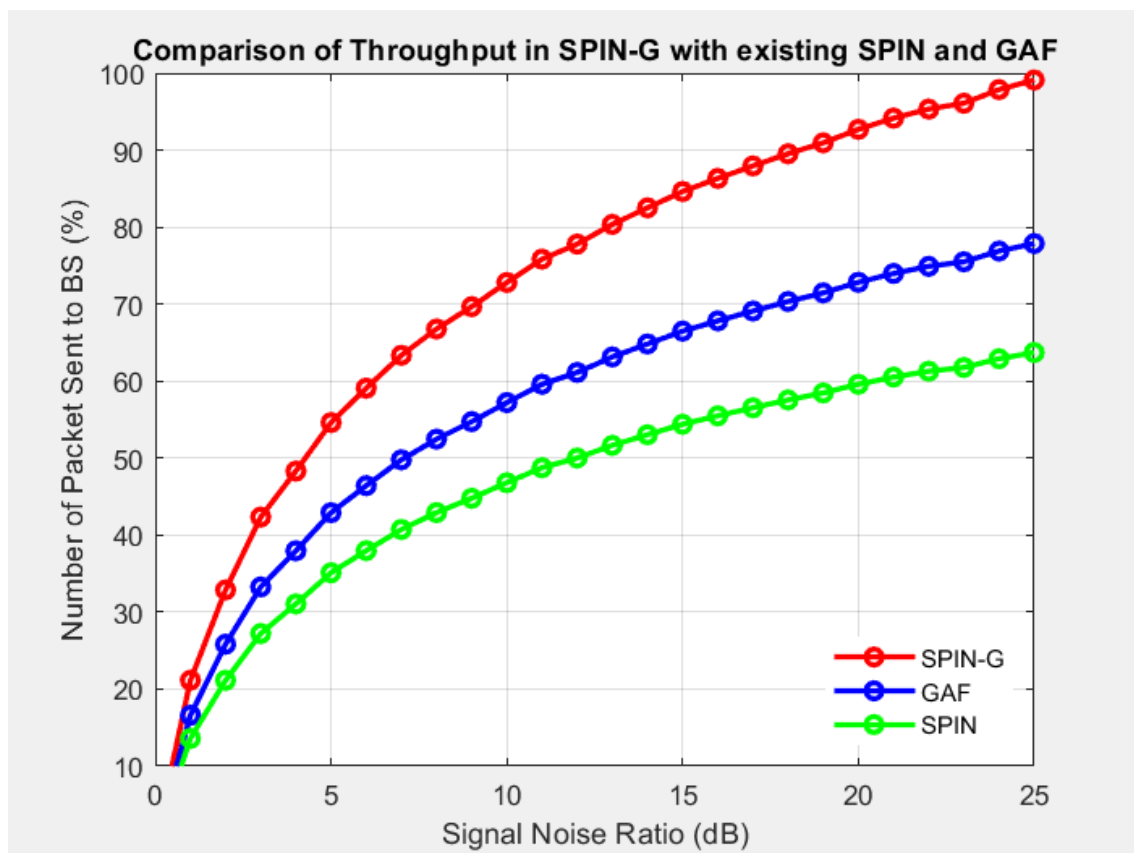


Figure 26: Using packet sent vs. Signal to Noise ratio to assess QoS in WSNs.

In Table 10, an assessment of three protocols, namely SPIN, GAF, and SPIN-G, is showcased. The evaluation is based on the bit error rate in relation to the signal-to-noise ratio. The simulation results, as shown in Figure 27, reveal that the proposed SPIN-G protocol exhibits fewer errors compared to both SPIN and GAF. This outcome signifies the superior performance of the SPIN-G protocol in terms of delivering packets.

Table 10: Comparing the results for GAF, SPIN, and SPIN-G protocols by evaluating the packets sent to the base station against the number of destinations.

Protocols	Bit Error Rate	Signal Noise Ratio (dB)					
		2dB	4dB	6dB	8dB	10dB	12dB
GAF		0.01126	0.007501	0.004576	0.0025	0.001191	0.0004777
SPIN		0.05628	0.03751	0.02288	0.0125	0.005954	0.002388
SPIN-G		0.005628	0.003751	0.002288	0.00125	0.0005954	0.0002388

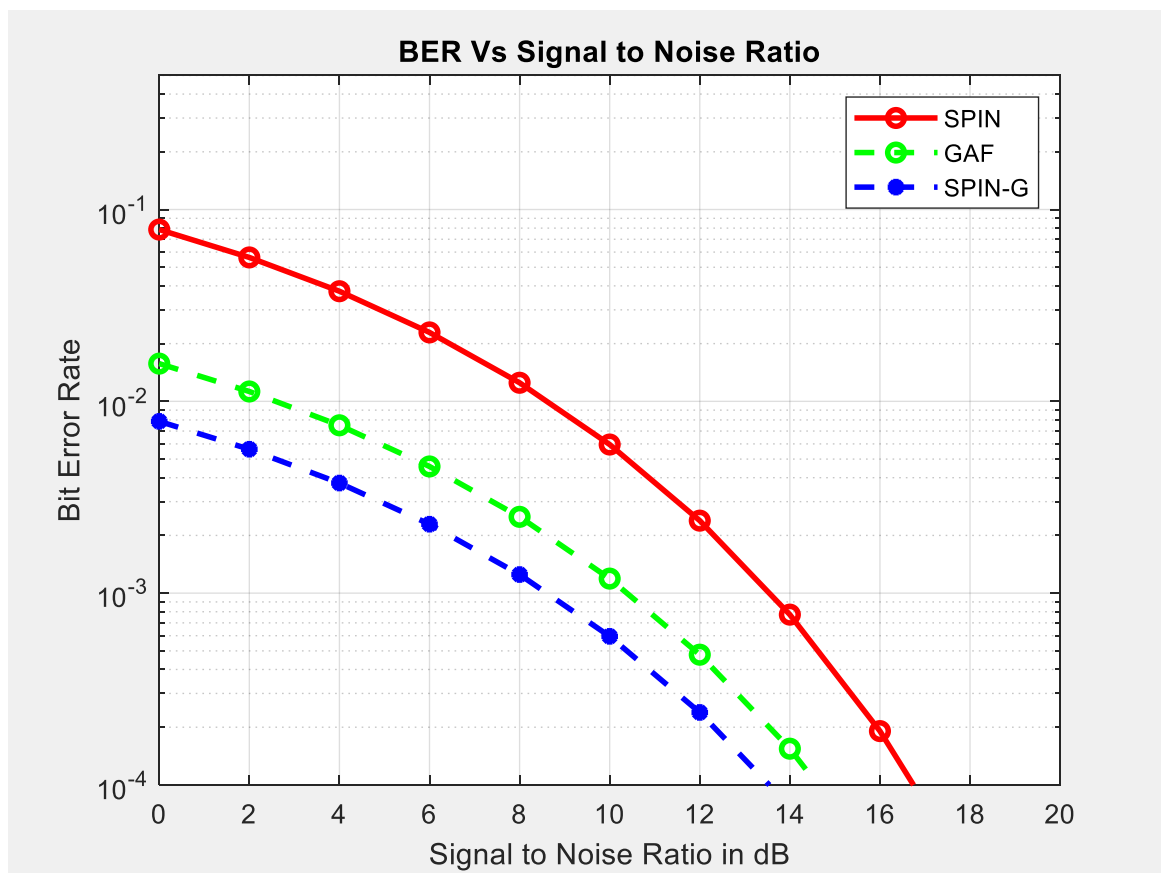


Figure 27: The performance evaluation of signal noise ratio against the bit error rate.

Table 11: This table shows the overall performance of three key parameters used in checking QoS.

Protocols	Overall Performance for Each Metric		
	Packet Loss (%)	Packet Delivery Ratio (%)	Latency (ms)
	25 Nodes	25Nodes	25 Nodes
GAF	13.97%	77.86%	0.000278ms
SPIN	16.77%	63.71%	0.000417ms
SPIN-G	11.18%	99.1%	0.0002085ms

4.6 Conclusion

In conclusion, this chapter has presented an analysis of the implementation of the SPIN-G protocol and the modified Dijkstra algorithm, highlighting their significant contributions to improving the quality of service (QoS) in wireless sensor networks (WSNs). The evaluation of SPIN's performance using various parameters has demonstrated its effectiveness in ensuring QoS within the protocol. Additionally, the integration of the GAF data delivery mechanism with SPIN has enhanced the mode of data dissemination, resulting in notable improvements in latency, throughput, and signal quality.

Traditionally, the broadcast mechanism in the SPIN topology posed challenges such as packet loss and delay. However, by transforming the broadcast mechanism into a data delivery scheme, the number of successfully delivered packets has increased significantly, with a remarkable 99.1% successful packet delivery rate at 25dB. Furthermore, the implementation of the modified Dijkstra algorithm has facilitated the creation of multiple paths with minimal latency and a modest packet loss rate of 11.18% when using 25 sensor nodes over the SPIN and GAF protocols.

This research emphasizes the crucial role of ensuring QoS in WSNs to achieve reliable and efficient data transmission. By considering important parameters such as network resources, delay time, and signal quality, and by designing protocols that incorporate QoS features, WSNs can effectively meet the requirements of diverse applications. The findings presented in this chapter provide valuable insights for future research and development in the field of WSNs, paving the way for further advancements in QoS-enabled protocols and enhancing the overall performance of WSNs in various application scenarios.

Chapter 5

5.0 Conclusion

In conclusion, the quality of service (QoS) is a crucial factor in the performance of wireless sensor networks (WSNs) and their protocols. QoS refers to a set of services that must be adhered to in order to meet specific system requirements, and its evaluation relies on various performance metrics aligned with desired objectives (Asif et al., 2017). In the case of the SPIN protocol, the broadcast mechanism has been identified as a cause of delays, collisions, and packet loss in WSNs.

This research has accomplished it aimed to enhance the efficiency of data delivery within the SPIN protocol by incorporating the data delivery mechanism of the GAF. To evaluate the QoS, performance parameters such as packet loss, throughput, signal noise ratio, and latency were implemented. The Modified Dijkstra algorithm, derived from the existing Dijkstra shortest algorithm for WSNs, was successfully employed in this study.

Routing plays a crucial role in ensuring successful packet delivery, and the research focused on mapping all sensor nodes into a grid and calculating the number of hops from the source to the destination. This approach improved the packet transmission rate in the SPIN-G protocol when merged with the Modified Dijkstra algorithm in the clustered grid topology.

The primary objective of this research was to address data delivery issues in the SPIN protocol, aiming to reduce latency and packet loss while enhancing the overall QoS in WSNs. As a result, the SPIN-G protocol was developed, which combined the strengths of the SPIN protocol and the Modified Dijkstra algorithm. This integration resulted in improved performance and QoS for data delivery in WSNs.

In conclusion, the research successfully addressed the challenges related to data delivery in the SPIN protocol, leading to the development of the SPIN-G protocol. By enhancing the efficiency of data transmission and reducing latency and packet loss, the proposed protocol offers an improved QoS for wireless sensor networks. These findings contribute to the ongoing efforts in optimizing protocols for WSNs, ultimately leading to better performance and reliability in SPIN-G protocol.

In conclusion, this research work has focused on addressing the challenges of problem-solving and improving the performance of the SPIN protocol in wireless sensor networks (WSNs). By integrating the modified Dijkstra algorithm into the SPIN-G protocol, significant enhancements have been achieved in terms of quality of service (QoS) and data delivery efficiency. The implementation of the modified Dijkstra algorithm has resulted in notable improvements in the data delivery rate, ensuring that packets are successfully transmitted and received within the network. This enhancement has directly contributed to minimizing latency, thereby optimizing the overall performance of WSNs.

It is important to recognize that problem-solving and continuous improvement are essential aspects of emerging and existing technologies. This research has successfully addressed the existing loopholes in the SPIN protocol and has demonstrated the potential for enhancing its performance in WSNs. Moving forward, further research and development are warranted to explore additional ways to improve the performance of WSNs. This may involve testing the modified Dijkstra algorithm on other protocols, evaluating its effectiveness in various network topologies, and investigating factors such as network lifetime and power consumption.

By continually striving for improvement and innovation, researchers can overcome challenges, enhance technology, and pave the way for more efficient and reliable wireless sensor networks, thus unlocking their full potential in a wide range of applications.

5.1 Future Improvements

In future research, there are several areas that should be explored to further enhance the implementation and performance of the proposed SPIN-G protocol in wireless sensor networks (WSNs). To begin with, it is important to conduct tests and evaluations of the proposed SPIN-G protocol with other new and existing protocols in WSNs, utilizing a diverse range of network topologies. This will allow for a comprehensive assessment of the protocol's applicability, effectiveness, and potential benefits in different network scenarios. By comparing its performance against alternative protocols, researchers can gain insights into its strengths and limitations, enabling the development of more efficient and reliable WSN solutions.

Moreover, while the implementation of SPIN-G was tested using a clustered grid topology with 50 sensor nodes, future work should consider evaluating its performance across various topologies. This would involve deploying larger numbers of sensor nodes and exploring different network configurations, such as hierarchical, mesh, or tree topologies. Evaluating SPIN-G's performance under these conditions will provide a broader understanding of its scalability, adaptability, and ability to meet QoS requirements in diverse WSN deployments.

In terms of network lifetime, it is essential to investigate and compare the duration of sensors in SPIN-G with and without the modified Dijkstra algorithm. This analysis would shed light on how the algorithm impacts the energy efficiency of the network, allowing for optimization strategies to be devised. By understanding the trade-offs between network lifetime and performance, researchers can design WSNs that achieve both prolonged operation and high-quality data transmission.

Additionally, the power consumption rate should be thoroughly evaluated in future research. This aspect, which was not addressed in the current study, represents a significant factor in the overall efficiency of WSNs. Analyzing the power consumption implications of the SPIN-G protocol, including the modified Dijkstra algorithm, would enable researchers to identify areas for improvement, develop energy-efficient mechanisms, and ultimately extend the lifespan of sensor nodes.

Finally, addressing transmission failures caused by dead nodes is another critical area for future investigation. Developing mechanisms or protocols that can detect and handle such failures would enhance the reliability and resilience of the network. By implementing strategies such as redundancy, node replacement, or dynamic routing, researchers can ensure that data packets are reliably delivered even in the presence of malfunctioning or non-operational nodes.

In conclusion, future research in the field of WSNs should focus on testing the SPIN-G protocol with various protocols and network topologies, evaluating network lifetime and power consumption, and developing mechanisms to mitigate transmission failures. By addressing these aspects, researchers can make significant advancements in improving the performance, efficiency, and reliability of WSNs, contributing to the development of more robust and effective solutions for various applications.

References

1. Tiwari, P., Saxena, V.P., Mishra, R.G. and Bhavsar, D., "Wireless Sensor Networks: Introduction, Advantages, Applications, and Research Challenges" *HCTL Open International Journal of Technology Innovations and Research (IJTIR)*, 2015, pp. 4-8.
2. P. Walia and A. Mehta, "Energy Efficient Geographic Adaptive Fidelity in Wireless Sensor Networks", *IOSR Journal of Computer Engineering*, vol. 17, no. 2278-0661, pp. PP 46-55, 2015.
3. M. Ali and R. Sai Kumar, 'Real-Time Support and Energy Efficiency in Wireless Sensor Networks', *Dissertation, Höskolan i Halmstad/Sektionen för Informationsvetenskap, Data- och Elektroteknik (IDE)*, 2008.
4. M. A. Matin and M. M. Islam, "Overview of wireless sensor network," *Wireless Sensor Networks - Technology and Protocols*, 2012.
5. S. Roychowdhury and C. Patra, "Geographic adaptive fidelity and geographic energy-aware routing in ad hoc routing", *IJCCT*, vol. 1, no. 2, 3, 4, p. 3, 2010.
6. D. Chen and P. K. Varshney, "QoS support in wireless sensor networks": *a survey*. *In International conference on wireless networks*, vol. Vol. 233, pp. 1–7, 2004.
7. S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.
8. M. BenSaleh, R. Saida, Y. Kacem and M. Abid, "Wireless Sensor Network Design Methodologies: A Survey", *Journal of Sensors*, vol. 2020, pp. 1-13, 2020.
9. D. Xiao, M. Wei, and Y. Zhou, "Secure-SPIN: Secure Sensor Protocol for Information via Negotiation for Wireless Sensor Networks", *IEEE*, p. 1, 2006.

10. M. Umale and S. D. Markande, "Energy-efficient routing algorithm on the target tracking in Wireless Sensor Network," *2015 International Conference on Information Processing (ICIP)*, Pune, 2015, pp. 176-181, DOI: 10.1109/INFOP.2015.7489373.
11. P. Tyagi, S. Jain, "Comparative Study of Routing Protocols in Wireless Sensor Network", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 2, Issue 9, September 2012.
12. D. Yuan, "Improving Quality of Service in Wireless Sensor Networks for Industrial Automation," PhD diss., Technische Universität Darmstadt, 2016.
13. R. KalaiMagal and M. Revathy "A survey on wireless sensor network protocols", *IJSET* vol.1, no 6, p. 12, 2014.
14. Garlapati, N. and Mohammed, A.A., "Evaluation of Power Conservation Algorithms in Industrial Wireless Sensor Networks", p. 19, 2012.
15. N. Shabbir and S. Hassan, "Routing Protocols for Wireless Sensor Networks (WSNs)", *Wireless Sensor Networks - Insights and Innovations*, 2017.
16. M. Javaid, "Wireless Sensor Networks: Software Architecture", *SSRN Electronic Journal*, 2014.
17. G. Sahar, K. A. Bakar, S. Rahim, N. A. Khani, and T. Bibi, "Recent advancement of data-driven models in wireless sensor networks: A survey," *Technologies*, vol. 9, no. 4, p. 76, 2021.
18. K. Zhao and L. Ge, "A Survey on the Internet of Things Security", *2013 Ninth International Conference on Computational Intelligence and Security*, 2013.
19. P. Walia and A. Mehta, "Energy Efficient Geographic Adaptive Fidelity in Wireless Sensor Networks", *IOSR Journal of Computer Engineering*, vol. 17, no. 2278-0661, pp. PP 46-55, 2015.

20. J. Lloret, M. Garcia, D. Bri, and J. Diaz, "A Cluster-Based Architecture to Structure the Topology of Parallel Wireless Sensor Networks," *Sensors*, vol. 9, no. 12, pp. 10513–10544, Dec. 2009.
21. A. M. Zungeru. "Energy-efficient routing algorithms based on swarm intelligence for wireless sensor networks." Ph.D. diss., University of Nottingham, 2013.
22. J. E. Mbowe and G. S. Oreku, "Quality of service in Wireless Sensor Networks," *Wireless Sensor Network*, vol. 06, no. 02, pp. 19–26, 2014.
23. A. Kochhar, P. Kaur, P. Singh, and S. Sharma, "Protocols for wireless sensor networks: A survey," *Journal of Telecommunications and Information Technology*, vol. 1, no. 2018, pp. 77–87, 2018.
24. O. Said, "Performance evaluation of WSN management system for QoS guarantee", *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, 2015. Available: 10.1186/s13638-015-0449-4.
25. C. Cirstea, M. Cernaianu, and A. Gontean, "Packet loss analysis in wireless sensor networks routing protocols," *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, 2012.
26. Y. L. Lee, J. Loo, and T. C. Chuah, "Modeling and Performance Evaluation of Resource Allocation for LTE Femtocell Networks," *Modeling and Simulation of Computer Networks and Systems*, pp. 683–716, 2015.
27. J. Grover, M. Sharma, and Shikha, "Reliable SPIN in Wireless Sensor Network," *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, 2014, pp. 1-6, DOI: 10.1109/ICRITO.2014.7014694.

28. M. Asif, S. Khan, R. Ahmad, M. Sohail, and D. Singh, "Quality of service of routing protocols in Wireless Sensor Networks: A Review," *IEEE Access*, vol. 5, pp. 1846–1871, 2017.
29. Z. Rehena, S. Roy and N. Mukherjee, "A modified SPIN for wireless sensor networks," 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), 2011, pp. 1-4, DOI: 10.1109/COMSNETS.2011.5716469.
30. D. S. Deif and Y. Gadallah, "Classification of Wireless Sensor Networks Deployment Techniques," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 834-855, Second Quarter 2014, DOI: 10.1109/SURV.2013.091213.00018.
31. O. Younis, M. Krunz and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," in *IEEE Network*, vol. 20, no. 3, pp. 20-25, May-June 2006, DOI: 10.1109/MNET.2006.1637928.
32. S. D. Kadu and V. S. Deshpande, "Handeling Throughput in wireless sensor network," 2012 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 2012, pp. 1-4, DOI: 10.1109/ICCIC.2012.6510281.
33. S. Sharma, R. K. Bansal, and S. Bansal, "Issues and Challenges in Wireless Sensor Networks," 2013 International Conference on Machine Intelligence and Research Advancement, Katra, India, 2013, pp. 58-62, DOI: 10.1109/ICMIRA.2013.18.
34. M. A. Rahman, A. E. Saddik, and W. Gueaieb, "Wireless Sensor Network Transport Layer: State of the art," *Lecture Notes Electrical Engineering*, pp. 221–245.
35. S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," in *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 443-461, Third Quarter 2011, DOI: 10.1109/SURV.2011.060710.00094.
36. J. E. Mbowe and G. S. Oreku, "Quality of service in Wireless Sensor Networks," *Wireless Sensor Network*, vol. 06, no. 02, pp. 19–26, 2014.

37. K. Sharma, N. Mittal, and P. Rathi, "Performance analysis of flooding and spin in wireless sensor networks," *International Journal of Future Generation Communication and Networking*, vol. 7, no. 3, pp. 25–36, 2014.
38. V. Soni and D. K. Mallick, "A novel scheme to minimize hop count for GAF in Wireless Sensor Networks: Two-level GAF," *Journal of Computer Networks and Communications*, vol. 2015, pp. 1–9, 2015.
39. H. Oudani, J. Laassiri, S.-ddine Krit, and L. El Maimouni, "Comparative study and simulation of flat and hierarchical routing protocols for Wireless Sensor Network," *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016.
40. P. Khan, G. Konar, and N. Chakraborty, "Modification of Floyd-Warshall's algorithm for Shortest Path routing in wireless sensor networks," *2014 Annual IEEE India Conference (INDICON)*, Pune, India, 2014, pp. 1-6, DOI: 10.1109/INDICON.2014.7030504.
41. K. M. Modieginnyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for Efficient Network Management: A survey," *Computers & Electrical Engineering*, vol. 66, pp. 274–287, 2018.
42. M. Salamath and S. R., "A survey on Spin Protocol in Wireless Sensor Network," *International Journal of Science and Research (IJSR)*, vol. 5, no. 3, pp. 1463–1466, 2016.
43. L. Pavithra, M. Abdullah, S. P. Prakash, S. Karthick, B. Ragavi, and V. D. Nandhini, "Wireless Sensor Networks: A review on sensor deployment and routing protocols for different application," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012052, 2021.
44. A. Kaur, G. Singh, and J. Kaur, "SPIN: A Data Centric Protocol For Wireless Sensor Networks," *International Journal of Engineering Research & Technology (IJERT)*, vol. Vol. 2, no. Issue 3, pp. 1–9, Mar. 2013.
45. A. Koubaa, R. Severino, M. Á. Alves, and E. Tovar, "Improving quality-of-service in wireless sensor networks by mitigating 'hidden-node collisions,'" *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 299–313, 2009.
46. W.-S. Jang and W. M. Healy, "Assessment of performance metrics for use of WSNs in buildings," *Proceedings of the 2009 International Symposium on Automation and Robotics in Construction (ISARC 2009)*, 2009.
47. D. Chauhan, R. Asthana, V. Gupta, and R. Kumar, "Spin based Hybrid Multi Hop Clustering Scheme," *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015.

48. G. Patel and P. N. M. Raja, "Routing Protocols to provide Quality of Service in Wireless Sensor Networks," *International Journal of Research in Advent Technology*, vol. 2, pp. 1–9, Feb. 2014.
49. S. Abbasian Dehkordi, K. Farajzadeh, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, and M. Abbasian Dehkordi, "A survey on data aggregation techniques in IOT sensor networks," *Wireless Networks*, vol. 26, no. 2, pp. 1243–1263, 2019.
50. P. Khan, G. Konar, and N. Chakraborty, "Modification of Floyd-Warshall's algorithm for shortest path routing in wireless sensor networks," *2014 Annual IEEE India Conference (INDICON)*, 2014.
51. L. F. W. van Hoesel, A. Tüysüz Erman, A. Dilo, and P. J. M. Havinga, "Geo-casting of queries combined with coverage area reporting for Wireless Sensor Networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 104–123, 2013.
52. D. Natarajan and A. P. Rajendran, "AOLSR: Hybrid ad hoc routing protocol based on a modified Dijkstra's algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, 2014. doi:10.1186/1687-1499-2014-90