Proposed Approach to Increase the Reliability of Wireless Sensor Networks through Dijkstra's Algorithm

Yaqin Jasim Abd Hashim[1,*], *Saad* Talib Hasson[2]

[1]Department of Information Networks, University of Babylon, Iraq
[2]Department of Information Networks, University of Babylon, Iraq

Abstract. This study aims to improve the performance of wireless sensor networks (WSNs) using Dijkstra's algorithm. WSNs are vital in applications like environmental monitoring and industrial processes, requiring efficient data flow and power management to extend their operational life. Dijkstra's algorithm is implemented to find the shortest paths between nodes, enhancing energy efficiency, network longevity, and data access speed at minimal cost. Key nodes are selected based on energy levels and proximity to key locations to facilitate data collection and transmission to the main hub, reducing the workload on other sensors and conserving energy. The algorithm also identifies alternative routes if multiple sensors share the same shortest path, reducing congestion, data loss, and interference. Extensive testing shows that Dijkstra's algorithm significantly improves sensor reliability, and data transmission speed, and reduces message loss compared to unregulated operation. This research confirms Dijkstra's algorithm's effectiveness in enhancing WSN functionality, particularly in energy conservation and sustained long-term performance. Implementing Dijkstra's algorithm enables WSNs to operate more efficiently and effectively, meeting energy conservation requirements while ensuring smooth long-term operation.

1 Introduction

Wireless Sensor Networks (WSNs) face challenges due to limited energy resources, requiring efficient management strategies to extend network lifespan. Dijkstra's algorithm offers a solution by optimizing routing, determining the shortest paths for data transmission, and reducing energy consumption. This also reduces congestion and data loss. However, its impact on performance metrics like reliability and throughput must be understood. This paper examines Dijkstra's algorithm's effectiveness in enhancing energy efficiency in WSNs and its practical implications for real-world use.

---

* *Yaqin* Jasim Abd Hashim: yaqeenjasima.net@student.uobabylon.edu.iq

2 Revie of Related Works

Akhilesh Panchal et al showed that a Wireless Sensor Network (WSN) typically comprises hundreds or thousands of nodes distributed manually or randomly across a designated area for environmental monitoring and tracking purposes. These nodes collect data from the environment, convert it into a suitable format, and transmit it to selected destinations. Energy conservation is a primary concern in WSNs, with clustering and packet routing playing crucial roles. [5]. Yingpeng Lin et al compared Dijkstra's and A* algorithms in an automated vehicle path planning context. The research provides a detailed comparative analysis, highlighting the strengths and weaknesses of each algorithm. The findings indicate that while A* offers faster convergence in certain scenarios, Dijkstra's algorithm remains robust and reliable for deterministic pathfinding. This reliability is particularly advantageous for WSNs, where predictable and energy-efficient routing is critical. [3]. Sabir Hossain et al demonstrated a prototype model using Dijkstra's algorithm to locate the nearest blood donors. The study showcases the algorithm's effectiveness in solving shortest-path problems in practical applications. Such real-world implementations underscore the potential of Dijkstra's algorithm in various WSN applications, such as locating the nearest sensor nodes for efficient data aggregation and transmission [10] Helong Wang et al Published in Ocean Engineering), this research proposed a three-dimensional version of Dijkstra's algorithm for optimal ship routing. The study introduces multi-objective optimization to account for many factors such as fuel consumption, travel time, and safety. This approach demonstrates the flexibility of Dijkstra's algorithm in handling complex routing scenarios, which can be translated to multi-hop routing in WSNs to optimize energy consumption and network longevity. [6] Given the limited power supply of wireless nodes, energy consumption is a primary design consideration, as it directly impacts the node's lifespan. Since it's often impractical to replace energy sources in these nodes, their longevity heavily relies on battery life. To extends the network's lifespan, various clustering algorithms have been proposed to optimize energy usage. This paper discusses several definitions of routing protocol that are aimed at conserving resources and prolonging the existence of WSNs. Despite advancements, WSN applications, including communication architectures, security, and management, still face unresolved challenges. By addressing these issues, they aim to bridge the gap between technology and implementation, thereby advancing the capabilities and efficiency of WSNs. Amhmed Bhih et al, highlighted that Wireless sensor networks consist of numerous sensor nodes capable of sensing, collecting, and processing data within the physical environment. These nodes are typically powered by batteries, which impose limitations on the network's lifespan. To address energy constraints and prolong network operation, sophisticated techniques are required for WSN deployment. [7] Wang Shu-Xi et al Improved Dijkstra's Shortest Path Algorithm and Its Application – Science Direct, this study discusses improvements to the classical Dijkstra algorithm and its applications in various fields. [8].

3 Methodology

Wireless Sensor Networks (WSNs) consist of small, battery-powered sensor nodes that collect and transmit environmental data wirelessly to a central station. These nodes are often used in remote areas for applications like environmental monitoring and smart cities. The proposed methodology implements Dijkstra's algorithm to optimize data transmission by finding the shortest path between nodes, differing from clustering approaches like LEACH. Implementation involves the following steps:

1. **Initialization**: Each sensor node is initialized with specific parameters, and source nodes are identified by their input IDs and output ports.

2. **Shortest Path Calculation**: Dijkstra's algorithm is used to compute the shortest path from each source node to all other nodes, minimizing transmission delay and optimizing energy use.

3. **Alternative Path Calculation**: To avoid congestion when multiple source nodes share the same path, an alternative shortest path is calculated for one of the source nodes.

4. **Reliability Assessment**: Network reliability is evaluated based on the shortest paths, considering factors like path length, data transmission rate, and potential conflicts.

This model aims to enhance data transmission efficiency and network reliability in WSNs using Dijkstra's algorithm, thereby improving overall network performance and Quality of Service (QoS) parameters. As we can see the main concept of Dijkstra’s algorithm in Figure 1.

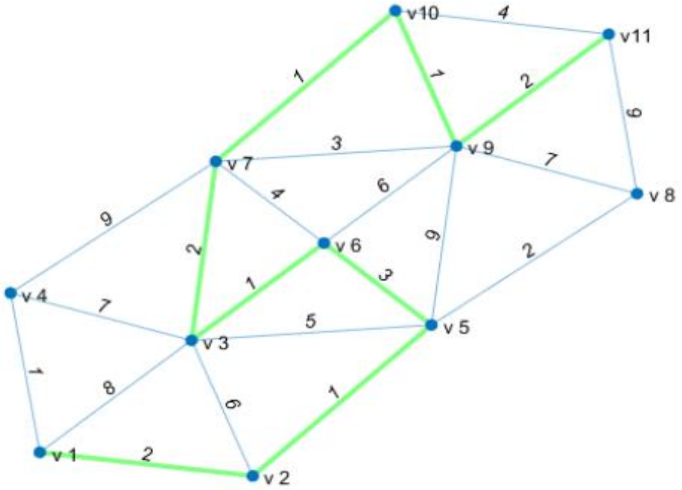


Fig. 1. Dijkstra’s Algorithm Topology

The proposed methodology focuses on evaluating communication protocols in WSNs, with an emphasis on using Dijkstra's algorithm to determine the shortest path for direct communication between nodes and the sink node. This serves as a baseline for evaluating key performance metrics such as throughput and packet delivery ratio (PDR). Comparative analysis provides insights into how path optimization impacts network efficiency and highlights Dijkstra's algorithm's effectiveness in improving WSN QoS parameters.

3.1 Origin of Dijkstra's Algorithm

Dijkstra's algorithm was introduced in 1956 by Dutch computer scientist Edsger W. Dijkstra to address the need for efficiently calculating the shortest path in telecommunication networks. The algorithm divides nodes into two sets: those with known shortest paths and those without. It iteratively selects the node with the shortest current distance from the source, updating neighboring nodes' distances until the shortest paths for all nodes are determined or no more reachable nodes remain. The simple steps of running Dijkstra's algorithm can be expressed as follows:

1. Start with initialization by setting the distance of the start node to 0 and the distance of all other nodes to infinity (indicating the shortest path that has not yet been determined). Mark the start node as the current node.
2. For the current node, compute the direct distance from the start node to it and update the distances to its neighboring nodes (if the newly computed distance is smaller than the currently stored distance). This ensures that each node will have a currently known shortest path.
3. Mark the current node as visited.
4. Select the next unvisited node as the new current node. The distance to this node should be the smallest of the currently known shortest paths.
5. Repeat steps 2 through 4 until all nodes are marked as visited or no reachable path exists.
6. Eventually, reconstruct the global shortest path by finding the shortest path from the start node to the target node.

Dijkstra's algorithm is valued for its simplicity, ease of implementation, and versatility in both directed and undirected graphs with non-negative edge weights. It accurately determines the shortest path from a single source to all other nodes and is widely used in logistics, transportation, and telecommunications. However, it is limited to solving single-source shortest paths and requires knowledge of all distances from the source, which can be impractical in large networks. Additionally, it is more time- and memory-intensive than some other algorithms, making it less suitable for very large-scale applications. The WSN model incorporates key parameters that influence performance and reliability. The number of nodes affects network size, density, and energy consumption, with more nodes improving coverage but increasing traffic and energy use. The transmission range impacts connectivity and energy demand, where a wider range enhances coverage but consumes more energy, as shown in Table 1 the simulation parameters

Table 1. Simulation Parameters

Parameter	**Value**
Nodes-Number	50
Sink-no	1
Rang	1-100
Time-interval	0-5000
Low_T	10.0
High_T	30.0
S-P	2000.0

Here is a diagram that indicates the main steps of the proposed work as shown in the Figure.2

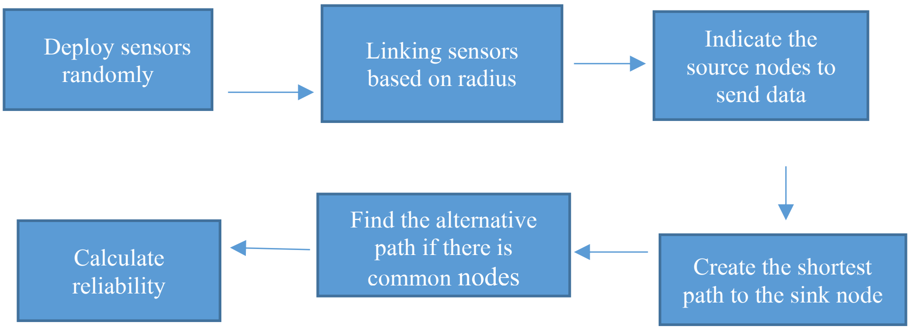


Fig. 2. General diagram of the proposed work

3.2 Equations of performance metrics

- Reliability are crucial Quality of Service (QoS) parameters in Wireless Sensor Networks (WSNs) that directly impact network performance and efficiency. Reliability, often denoted by R, represents the probability that a packet successfully reaches its destination within a given time frame. Mathematically, it is defined as the ratio of successful packet transmissions to the total number of attempted transmissions

$$R=P_S \backslash P_T \ldots\ldots\ldots\ldots\ldots (1)$$

Where:
Ps: Successful Transmissions
Pt: Total Number of Transmissions

A higher reliability indicates a more dependable network where data is consistently transmitted without loss

- Throughput, denoted by T, measures the rate at which data packets are successfully delivered from source to destination over the network. It is typically expressed in packets per second (pps) or bits per second (bps). Mathematically, throughput is calculated as:

$$T=T_S \backslash T_{total} \ldots\ldots\ldots\ldots (2)$$

Where:
Ts: Successful Transmissions
T total: Total Time

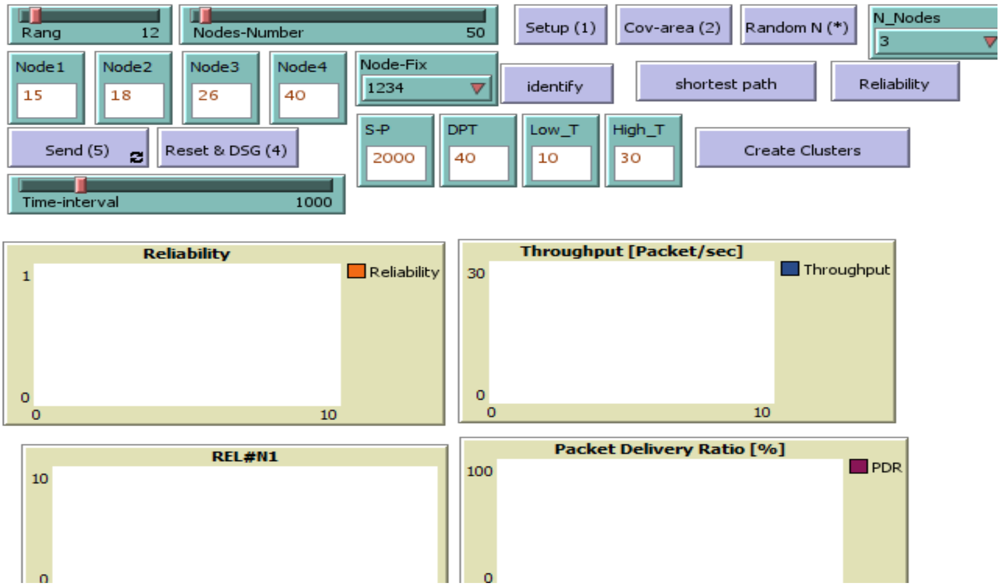
A higher throughput signifies a network's ability to handle a greater volume of data traffic efficiently

In Wireless Sensor Networks (WSNs), enhancing Quality of Service (QoS) parameters is crucial for efficient and reliable communication. High reliability ensures accurate and timely data transmission, minimizing information loss and errors. Improved throughput allows the network to handle higher data rates, facilitating prompt delivery of information. Maximizing

Packet Delivery Ratio (PDR) ensures that most data packets reach their destination, reducing retransmissions and boosting overall network efficiency.

4 Simulation and results

The simulation using NetLogo is crucial for modeling and analyzing Wireless Sensor Networks (WSNs). NetLogo's user-friendly interface and flexible programming environment facilitate the simulation of various scenarios, allowing detailed examination of node interactions and network dynamics. Its agent-based approach and real-time visualization capabilities enhance the analysis of network behavior and performance. By leveraging NetLogo, this study effectively explores WSN intricacies and evaluates the impact of different parameters and algorithms, as shown in Figure 3.



Rang 12
Nodes-Number 50
N_Nodes 3
Node1 15
Node2 18
Node3 26
Node4 40
Node-Fix 1234
S-P 2000
DPT 40
Low_T 10
High_T 30
Time-interval 1000

Fig. 3. Netlogo Simulation Tool

To create a wireless sensor network (WSN), sensors are randomly distributed within the network area. They are then connected based on a predefined communication range to establish a functional network. Source nodes are identified for calculating the shortest path to the sink node using Dijkstra's algorithm. In cases where sensors intersect or share the same path, alternative routes are computed to avoid congestion and ensure efficient data transmission.

Steps:

1. Distributing Sensors: Sensors are randomly distributed within a specified area, simulating real-world conditions. This approach creates a flexible network. Figure. 4 illustrates the deployment of sensors in the NetLogo simulation.

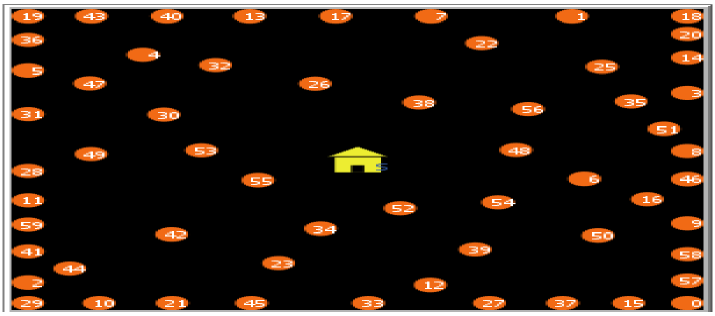


Fig. 4. Deploy sensors randomly

2. Indicating Communication Range: Define a communication range to determine the distance within which sensors can communicate directly. This range establishes whether two sensors can connect.

3. Connecting Sensors: Establish links between sensors within the defined communication range, forming a communication network that enables data exchange. Figure. 5 shows the sensor connections, and Figure. 6 displays the ID numbers of source nodes.

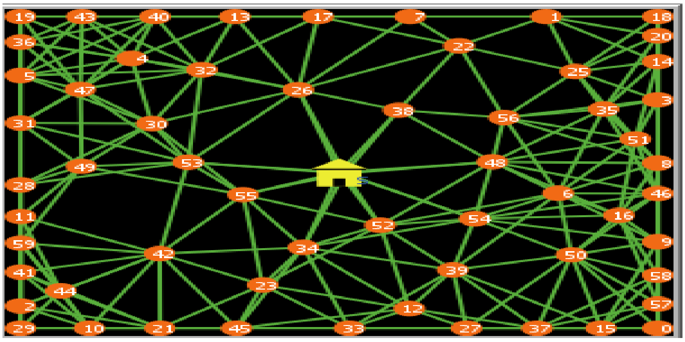


Fig. 5. Linking sensors to each other

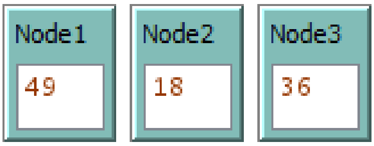

Node1
49
Node2
18
Node3
36

Fig. 6. Select source nodes

4. **Identifying Source Nodes:** Source nodes are designated to transmit data to the sink node, serving as the starting points for path calculations. Refer to Fig. 6.
5. **Calculating the Shortest Path Using Dijkstra's Algorithm:** Dijkstra's algorithm calculates the shortest path from the source nodes to the sink node, determining the optimal route based on distance or cost. Fig. 7 illustrates the shortest path.

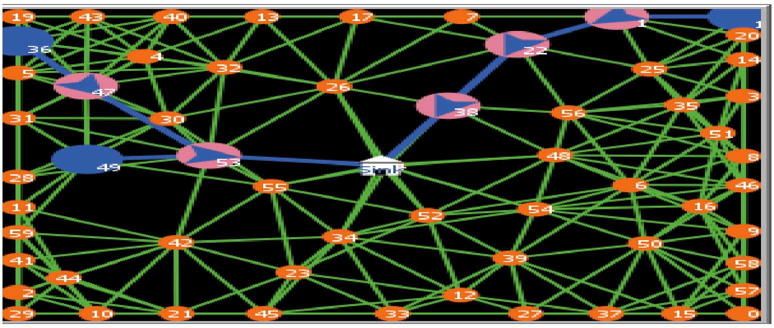


Fig.7. Indicates the shortest path

6. Calculating Alternative Paths: If there are intersections between sensors, an alternative path is calculated to ensure continuous communication. This helps improve network reliability and avoid potential failure points. Fig .8 shows the alternative path.

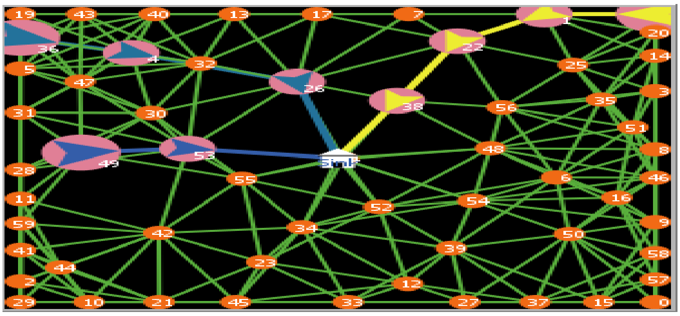


Fig. 8. indicates the alternative path

7. Sending messages: Send messages from source nodes to the sink node and calculate reliability for each source, an average of reliability, packet delivery ratio, throughput, error rate, number of success messages, and number of field messages as shown in Figure. 9.

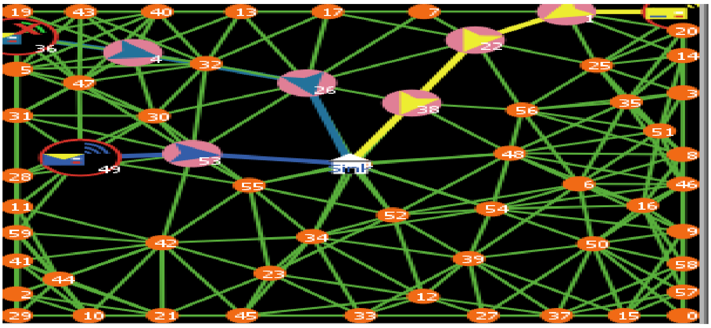


Fig. 9. Sending messages

8. Calculating metrics : Reliability, throughput, and PDR of the WSN Scenario as shown in Figure 10, Figure 11.

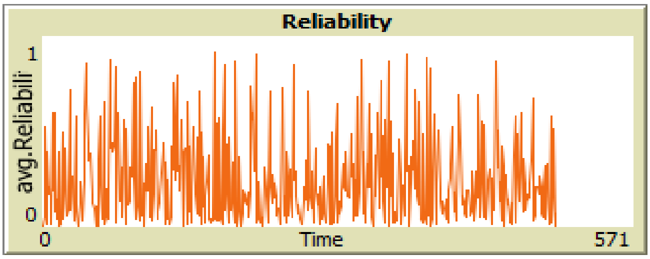
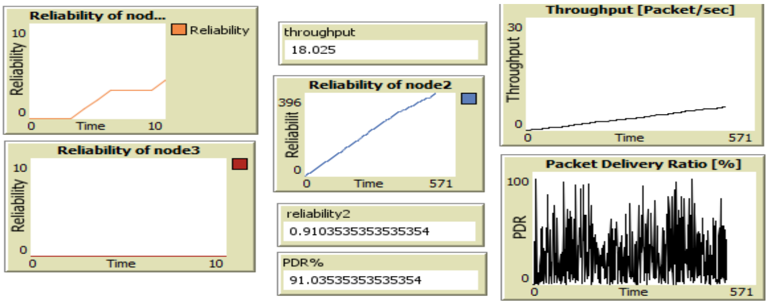


Fig. 10. Calculation of reliability



throughput
18.025

reliability2
0.9103535353535354

PDR%
91.03535353535354

Fig. 11. Calculation of other metrics

The results show that the network's reliability remains consistently high, even as the number of nodes increases. Although there's a slight decrease in reliability with more nodes, the values stay above 90%, indicative of a robust and well-functioning network. This analysis highlights the effectiveness of the network in maintaining reliability across different configurations. Figure 12. shows that the reliability decreases as the number of sensors increases due to the increased probability of errors that can occur with a higher volume of message transmissions.

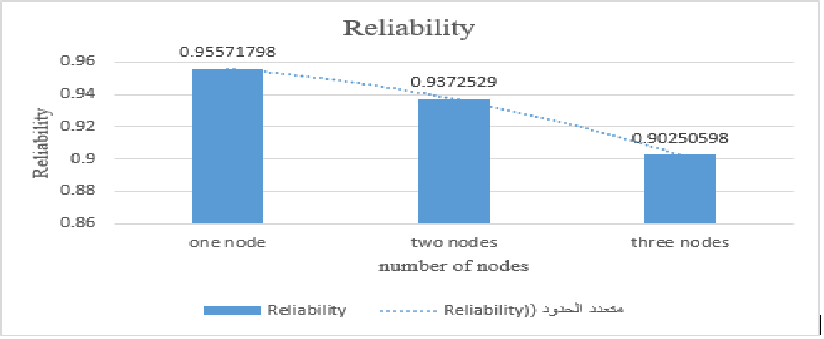


Fig. 12. The Reliability

5 Conclusions

Our research demonstrates that using Dijkstra's algorithm significantly enhances the reliability of message and data transmission in Wireless Sensor Networks (WSNs). The reliability of message delivery was recorded at 90%, indicating a substantial improvement. This enhancement is achieved without excessive energy consumption or message loss. By efficiently calculating the shortest paths, Dijkstra's algorithm ensures that data packets are transmitted more reliably and efficiently across the network. This underscores the importance of employing robust algorithms like Dijkstra's to achieve stable and effective network operations in WSN

References

1. Capdehourat, Germán, Paola Bermolen, Marcelo Fiori, Nicolás Frevenza, Federico Larroca, Gastón Morales, Claudina Rattaro, and Gianina Zunino. 2021. "Large-Scale 802.11 Wireless Networks Data Analysis Based on Graph Clustering." Wireless Personal Communications 1–29.
2. AliReza Naderloo, "Fuzzy-based cluster routing in wireless sensor network", Iran University of Medical Sciences, Tehran, Iran, 2023.
3. Yingpeng Lin "Dijkstra and A* Algorithms in Automated Vehicle Driving,* SPIE Digital Library, 2023.
4. Parwez, Md Salik, Danda B. Rawat, and Moses Garuba. 2017. "Big Data Analytics for User-Activity Analysis and User-Anomaly Detection in Mobile Wireless Network." IEEE Transactions on Industrial Informatics 13(4):2058–65.
5. Sabir Hossain "Research on Robot Path Planning Based on Dijkstra and Ant Colony", IEEE Xplore, 2022.
6. Chin lin "comparative Study Of Various Approaches Of Dijkstra Algorithm", IEEE Xplore, 2021..
7. Amhmed Bhih, "An Improved Leach Algorithm Based on Hierarchical Clustering Approach for Wireless Sensor Network Application", Department of Communication College of Electronic Technology (CET)-Tripoli Tripoli, Libya, 2022.
8. N. Meenakshi, "Efficient Communication in Wireless Sensor Networks Using Optimized Energy Efficient Engroove Leach Clustering Protocol", College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia, 2024.
9. Mali, Ganesh U., and D. K. Gautam. 2018. "Shortest Path Evaluation in Wireless Network Using Fuzzy Logic." Wireless Personal Communications 100(4):1393–1404.
10. Yang, Howard H., Chao Xu, Xijun Wang, Daquan Feng, and Tony Q. S. Quek. 2021. "Understanding Age of Information in Large-Scale Wireless Networks." IEEE Transactions on Wireless Communications.
11. Yilmaz, Onur, Sercan Demirci, Yagiz Kaymak, Serkan Ergun, and Ahmet Yildirim. 2012. "Shortest Hop Multipath Algorithm for Wireless Sensor Networks." Computers & Mathematics with
12. Akyildiz, Ian F., Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. 2002. "Wireless Sensor Networks: A Survey." Computer Networks 38(4):393–422.
13. Mali, Ganesh U., and D. K. Gautam. 2018. "Shortest Path Evaluation in Wireless Network Using Fuzzy Logic." Wireless Personal Communications 100(4):1393–1404.

14. SONAM LATA 1, SHABANA MEHFUZ 1, (Senior Member, IEEE), SHABANA UROOJ 2,3, (Senior Member, IEEE), AND FADWA ALROWAIS 4" Fuzzy Clustering Algorithm for Enhancing Reliability and Network Lifetime of Wireless Sensor Networks"2021
15. Faheem Khan 1, Shabir Ahmad 1, Hüseyin Gürüler 2, Gurcan Cetin 2, Taegkeun Whangbo 1 and Cheong-Ghil Kim 3, *" An Efficient and Reliable Algorithm for Wireless Sensor Network "2021
16. Jianfeng Yang, Ming Zhao & Jing Chen" ELS algorithm for estimating open source software reliability with masked data considering both fault detection and correction processes " 2021
17. Jianfeng Yang1,3, Jing Chen2, and Xebio Wang1" EM Algorithm for Estimating Reliability of MultiRelease Open Source Software Based on General Masked Data "2021
18. Bo Zhao, Jianfeng Yang*, Ming Zhao, Qi Li," Wireless sensor network reliability modelling based on masked data " 2015