

**DESIGN AND IMPLEMENTATION OF AN ENERGY EFFICIENT K-
BARRIER COVERAGE NETWORK FOR RURAL APPLICATIONS**

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Electronics Engineering

By
Vinay Chowdary

August 2019

SUPERVISOR(s)

**Dr. Mukul Kumar Gupta
Dr. Sushabhan Choudhury**



**Department of Electrical & Electronics Engineering
School of Engineering
University of Petroleum and Energy Studies
Dehradun – 248007: Uttarakhand**

DESIGN AND IMPLEMENTATION OF AN ENERGY EFFICIENT K-BARRIER COVERAGE NETWORK FOR RURAL APPLICATIONS

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Electronics Engineering

By
Vinay Chowdary
(SAP ID: 500034355)

August 2019

Internal Supervisor(s)

Dr. Mukul Kumar Gupta
Assistant Professor – Selection Grade
Department of Electrical & Electronics Engineering
University of Petroleum and Energy Studies

Dr. Sushabhan Choudhury
Professor
Department of Electrical & Electronics Engineering
University of Petroleum and Energy Studies



Department of Electrical & Electronics Engineering
School of Engineering
University of Petroleum and Energy Studies
Dehradun – 248007: Uttarakhand
August 2019

June 2020

DECLARATION

I declare that the thesis entitled “**Design and Implementation of an Energy Efficient k-barrier Coverage Network for Rural Applications**” has been prepared by me under the guidance Dr. Mukul Kumar Gupta, Assistant Professor – Selection Grade and Dr. Sushabhan Choudhury, Professor, of Department of Electrical & Electronics Engineering, University of Petroleum and Energy Studies. No part of this thesis has formed the basis for the award of any degree or fellowship previously.



Vinay Chowdary

Department of Electrical & Electronics Engineering,

University of Petroleum and Energy Studies,


Dehradun – 248007: Uttarakhand

DATE : 04-06-2020

CERTIFICATE

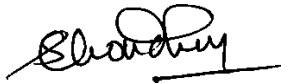
I/We certify that Vinay Chowdary has prepared his thesis entitled “**Design and Implementation of an Energy Efficient k-barrier Coverage Network for Rural Applications**”, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the Department of Electrical & Electronics Engineering, University of Petroleum & Energy Studies.

Internal Guides



Dr. Mukul Kumar Gupta

Assistant Professor – Selection Grade
Department of Electrical & Electronics Engineering
University of Petroleum and Energy Studies
Dehradun – 248007: Uttarakhand.



Dr. Sushabhan Choudhury

Professor
Department of Electrical & Electronics Engineering
University of Petroleum and Energy Studies
Dehradun – 248007: Uttarakhand.

DATE : 04-06-2020

ABSTRACT

Barrier coverage, also referred as *k*-barrier coverage, is the prominent research area in the field of coverage in wireless sensor networks. Barrier coverage is a special case of full coverage in wireless sensor networks. Full coverage of a network needs redundant and sequential deployment of sensors, which means that the area to be covered is small, whereas in general, wireless sensor networks are preferred in applications that are remote and inaccessible to humans. In all such applications, random deployment is preferred over sequential deployment. Barrier coverage uses less number of sensors than full coverage, still providing coverage, which is close to full coverage. In barrier coverage, it is essential that the value of $k \geq 1$, where k represents the number of barriers formed in the network. A barrier will be formed if there exists a path of connected sensors from one end of the network to the other end. Coverage provided by the network increases if the number of barriers formed in the network increases i.e., the value of k should be maximized. To maximize the number of barriers, techniques proposed in the literature have majorly focused on post deployment strategies such as use of mobile sensors or relocating sensors after initial deployment. This needs extra cost in terms of energy required, which reduces the operating lifetime of network. In this thesis, a novel, pre-deployment strategy based, Minimum Non-Overlap Radius Deployment Algorithm (MNORDA) is proposed where a minimum distance between coordinates of two nodes is maintained prior to deployment. With this, MNORDA guarantees no two nodes are deployed close to each other and also that no node will be deployed close to or beyond the edge of the network which helps in achieving uniform coverage of the network with less number of sensors than full coverage. MNORDA maximizes the value of k and enhances the operating lifetime of the network. The simulation results proves the superiority of proposed algorithm when compared to other existing state-of-art works. Hardware for rural applications is developed by forming barrier of sensor nodes extending the Wi-Fi to the nearest place and for detecting forest fire in the region of interest with the help of Internet of Things (IoT) and ZigBee with drones.

ACKNOWLEDGEMENTS

I, first and foremost, would like to express my deepest gratitude to Dr. Mukul Kumar Gupta, my supervisor and Dr. Sushabhan Choudhury, my co-supervisor, for encouraging me to take this research problem on Design and Implementation of an Energy Efficient k-barrier Coverage Network for Rural Applications. My supervisors has been a source of inspiration and monument of motivation to me throughout this work. Without them, this work would never have been completed.

I am thankful to Chancellor Dr. S. J. Chopra, Vice Chancellor Dr. Sunil Rai, Dean SOE Dr. Kamal Bansal, and Dr. Jitendra Kumar Pandey, Dean R&D for motivating me with space and environment in the serene campus of University of Petroleum & Energy Studies, Dehradun, in India to carry out this work. I would like to mention a special thanks to team R&D, UPES for providing constant guidance and support.

I should mention my gratitude to HoD, Electrical & Electronics Engineering, Dr. R Gowri who has always motivated me throughout the journey.

I would like to thank my colleagues Dr. Abhinav Sharma, Dr. Vivek Kaundal, and my students Mr. Udayveer Mittal, Mr. Thrilochan Sharma, Mr. Abhiram who have helped me while taking readings in outdoor locations and set up of the wireless network. I would like to thank to Dr. Arpit Jain and Mr. Deepak Kumar for helping me in editing of my thesis. Also, I am thankful to each and every faculty colleague of Electrical and Electronics Engineering department for their support and cooperation.

I would like to thanks Ms. Rakhi Ruhel, Mr. Sony Sandeep farmer and other CCE staff members for their help and numerous occasions.

Finally, I must express my very profound gratitude to my father Mr. Rajender Chowdary, mother Mrs. Shyama Chowdary, brother Mr. Vikash Chowdary and family, relatives-in-law for their moral support and numerous blessings.

Most importantly, I thank my wife Mrs. Sarika Chowdary and daughters Samriddhi Chowdary and Siya Chowdary for always being there with me and being a persistent source of encouragement.

Thank you.

Vinay Chowdary
University of Petroleum & Energy Studies

DEDICATED
TO

MY WIFE
&
DAUGHTERS

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES	PAGE NO.
1. Figure 1.1 Architecture of WSN	2
2. Figure 1.2 Full Coverage	4
3. Figure 1.3 Barrier Coverage	4
4. Figure 1.4. Types of Barrier Coverage	6
5. Figure 1.5 Techniques of Barrier Coverage	8
6. Figure 3.1 Nodes with their range circles	29
7. Figure 3.2. Network with $O_R = 0$, $V = 20$, $X_{max} = 75$, $Y_{max} = 45$, $X_{offset} = 1$, $Y_{offset} = 1$	33
8. Figure 3.3. Network with $O_R = 5$, $X_{offset} = 25$, $Y_{offset} = 1$	33
9. Figure 3.4 Example network with 10 nodes.	37
10. Figure 3.5 Network with 20 nodes and $O_R = 0$, $X_{offset} = Y_{offset} = 10$	43
11. Figure 3.6 Network with 20 nodes and $O_R = 5$, $X_{offset} = Y_{offset} = 10$	43
12. Figure 3.7 Network with two edge-disjoint paths.	45
13. Figure 3.8 Number of sensors required in traditional methods	45
14. Figure 3.9 Number of sensors required in MNORDA algorithm	46
15. Figure 3.10 Network for calculating shortest path	48
16. Figure 4.1 Processing software	50
17. Figure 4.2 Number of Barriers vs Number of Sensor Nodes	52
18. Figure 4.3 Network with 100 nodes and 18 barrier paths	53
19. Figure 4.4 Result showing 18 disjoint paths along with length and distance	53
20. Figure 4.5 Network with 500 nodes and 29 disjoint paths	54
21. Figure 4.6 Result showing 29 disjoint paths along with length and distance	54
22. Figure 4.7 Consumed Energy vs Number of Sensor Nodes	56
23. Figure 4.8 Result showing packets exchanged for network	

with 100 nodes	57
24. Figure 4.9 Termination time vs number of sensor nodes	58
25. Figure 4.10 Lifetime of barrier in weeks	60
26. Figure 4.11 Sensor Utilization vs Number of Sensor Nodes	61
27. Figure 4.12 Effect of OR on number of barriers	62
28. Figure 4.13 Sample result for MNORDA	63
29. Figure 4.14 Paths for node size of 900	65
30. Figure 4.15 Paths for node size of 700	65
31. Figure 5.1 Sensor node of a barrier	67
32. Figure 5.2 NodeMCU in STA mode	69
33. Figure 5.3 ESP8266 in AP mode	72
34. Figure 5.4 Set-up for Wi-Fi range extender	74
35. Figure 5.5 Wi-Fi Extender experiment with 3 nodes	75
36. Figure 5.6 Speed Test of Wi-Fi Range Extender	77
37. Figure 5.7 Architecture of MQTT	78
38. Figure 5.8 Configuration between MQTT box and cloudmqtt.com	79
39. Figure 5.9 Successful connection between MQTT box and cloudmqtt.com	80
40. Figure 5.10 Uploading DHT11 data to IoT Cloud	81
41. Figure 5.11 Set up of sensor nodes for forest fire detection	82
42. Figure 5.12 Experiment set up for multi-hop data transfer	83
43. Figure 5.13 Power consumption analysis of sensor node	84

LIST OF TABLES

TABLE	PAGE NO.
1. Table 2.1: Comparison of algorithm based approaches	20
2. Table 2.2: Summary of Literature Review	24
3. Table 4.1 Number of barriers for node size for all algorithms	51
4. Table 4.2 Total number of Packets ($\times 10000$) Exchanged	55
5. Table 4.3 Termination time in seconds for all algorithms	57
6. Table 4.4 Lifetime versus number of nodes for all algorithms	58
7. Table 4.4 Percentage of Sensor Utilization	60

LIST OF EXHIBITS

CHAPTER	PAGE NO.
1. Chapter 1 – Introduction	1
1.1 Introduction to WSN	1
1.2 Introduction to Coverage	3
1.3 Types of Barrier Coverage	5
1.4 Techniques of Barrier Coverage	6
1.5 Applications of Barrier Coverage	8
1.6 Limitations of Barrier Coverage	9
1.7 Introduction to Hardware	10
2. Chapter 2 – Review of Literature	12
2.1 Centralized Approach	12
2.2 Distributed Approach	13
2.3 Deployment Based Approach	15
2.4 Sensor Based Approach	16
2.5 Algorithm Based Approach	18
2.6 Summary of Literature Review	21
2.7 Conclusion of Literature Review	24
2.8 Objectives	25
3. Chapter 3 – Methodology and Algorithm Description	26
3.1 Methodology	26
3.2 Function in MNORDA Algorithm	27
3.3 Algorithm Description	30
3.4 Node Position Set	31
3.5 Find Neighbor Algorithm	35
3.6 Tell-Neighbor Algorithm	38
3.7 Generate Adjacency Matrix	39
3.8 Implementing Symmetry in Adjacency Matrix	40
3.9 Construct Barrier Paths	41
3.10 Identification of Isolated Nodes	46
3.11 Identification of Shortest Path in terms of Distance	

And Number of Nodes	47
4. Chapter 4 – Simulation Results	50
4.1 Simulation Result	51
4.2 Performance of MNORDA for different values of O_R	62
4.3 MNORDA Sample Result	63
5. Chapter 5 – Hardware Implementation	67
5.1 Hardware Components	67
5.2 Hardware of Applications of Barrier Coverage	68
5.3 Wi-Fi Range Extender for Rural Applications through Barrier Coverage	73
5.4 IoT based Forest Fire Detection System for Rural Applications	77
5.5 Forest Fire Detection through Multi-hop Data Transfer	82
5.6 Power Consumption Analysis of Sensor Node	83
6. Chapter 6 – Conclusion	85
6.1 Highlights of Proposed MNORDA Algorithm	87
6.2 Future Scope	87
REFERENCES	88

CHAPTER 1 - INTRODUCTION

The chapter describes the introduction of Wireless Sensor Networks (WSN), Barrier Coverage, types of coverage, types of barrier coverage, its applications and techniques to achieve barrier coverage. This chapter also describes the introduction of hardware developed to achieve barrier coverage.

1.1 Introduction to WSN

WSN is a constant companion for humankind since time immemorial. Over the last decades, application of WSN has seen an exponential growth. Although the application of wireless sensor networks is enormous, they are being increasingly used in civil applications such as health care, smart home, environmental monitoring, wireless meter reading, for providing guidance to group tours and in military applications such as detection of intrusion in an intelligent way, border monitoring system and rural applications such as precision agriculture, precision viticulture, green house environment monitoring and in post-disaster situations for human tracking especially in cases related to floods, major earthquakes, landslides in mountain regions and in pre-disaster situations such as forest-fire detection and monitoring.

Irrespective of application, the fundamental concerns that have to be considered in wireless sensor networks are listed as follows:

1. Deployment of these networks in hostile and unattended environments.
2. Limited energy resource in battery operated power conservative wireless network which has to be always kept operational by keeping it on
3. Cost effectiveness of the wireless sensor network.
4. Requirement of these networks to operate under ad-hoc condition that is when no infrastructure is available.

To address these concerns, the wireless sensor networks should therefore be able to:

1. Provide robustness in all weather conditions.
2. Have a very long lifetime.
3. Provide complete coverage of Region of Interest (RoI).
4. Operate under very less maintenance.

5. Function as an infrastructure less network.

Owing to their flexibility and scalability at affordable cost, Wireless Sensor Networks have always been considered as reference technology for detection, controlling and monitoring environmental tasks at a very large scale. WSN is a group of sensor nodes able to sense data or event which falls within its sensing range. The sensed data is transferred through multiple hops to the sink nodes or a gateway where it can be processed as shown in figure 1.1

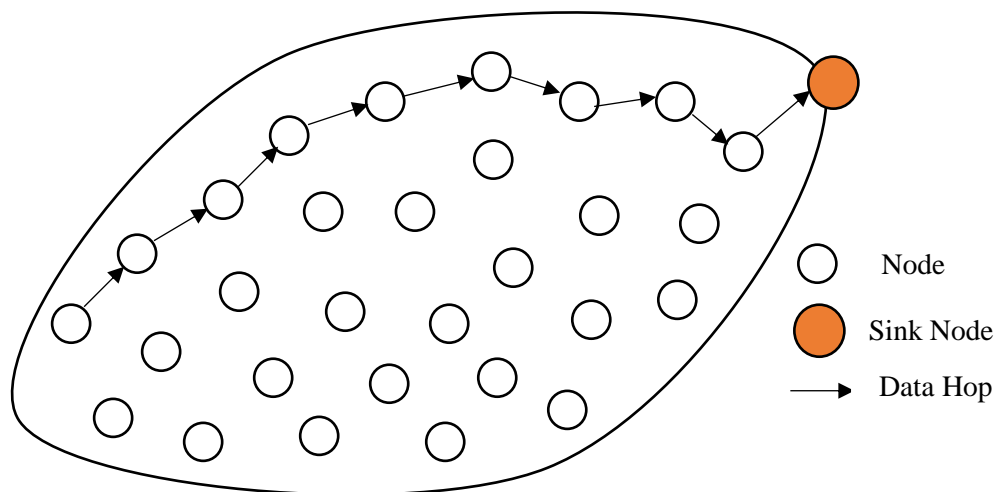


Figure 1.1 Architecture of WSN

Every node sends the sensed data to its neighbor node, if it is within its communication range. Because of which the data reaches the sink node. It is at this sink node that processing of data is performed for decision-making. Sink node has higher communication range, higher processing capability and a perpetual battery life. In general, every sensor node will have four different devices, which are:

1. **Sensing Device:** This device is responsible for sensing the event like temperature, humidity, light, etc., in the Region of Interest (RoI) where the node is deployed. This device will generate raw data, which may need a calibration before it can be delivered to sink node.
2. **Processing Device:** Processing device will mostly be a microcontroller which is used to process raw data received from sensing device. Processing device is also responsible for decision making depending on the value of sensed data.
3. **Transceiver Device:** Forwarding of data to neighboring hop is taken care of by the transceiver device. The communication range of sensing device depends on the power

of the transceiver device. Higher power will result in higher communication range but it will also affect the battery life of the device, as transmission power is directly proportional to battery consumption. Therefore there should be trade-off between required transmission power and battery life in the network

4. Power Device: One of the most important devices in the network is the power device. This device powers the other devices like sensing, processing and transceiver. Operating lifetime of whole network directly depends on the battery life of devices. Intelligent energy saving approach can be applied which helps in increasing the overall lifetime of the network.

1.2 Introduction to Coverage

In WSN, for measuring quality of service, coverage is the unanimously accepted metric. The objective in a WSN is primarily towards achieving maximum coverage and to have almost every point in the area of interest under the sensing range of minimum one or more than one sensor. Coverage provided by the WSN determines how efficiently the sensor nodes monitor the network. In WSN, ways of coverage fall under following categories: full and partial coverage. In the full coverage type, each point of the region should fall under the sensing range of at least one sensor and this requires additional and redundant sensor nodes where as in partial coverage redundant nodes are eliminated by covering only a portion of the entire region. This research is focused towards *k-barrier coverage* also called as *barrier coverage* (barrier coverage and k-barrier coverage will be considered as synonyms throughout the thesis), which is a special case of partial coverage where the coverage of the whole region is not the objective. As it is unnecessary to cover every point in the region, much fewer sensors are required in barrier coverage than in full coverage. Therefore, barrier coverage can act as a substitute for full coverage in cases where fewer sensors are required as it acts as an appropriate coverage model in WSN. Figure 1.2 and 1.3 gives an example network for full and barrier coverage respectively.

In full coverage shown in figure 1.2 every point in the area of interest falls under the sensing range of at least one sensor and therefore coverage gaps are almost nil. This needs sequential deployment of sensors to ensure that the network is completely covered. Sequential deployment is easy to be carried out when the size of the network is less and the area to be monitored is human friendly. WSN is mostly used for environments, which are not human friendly and to monitor area of larger dimensions.

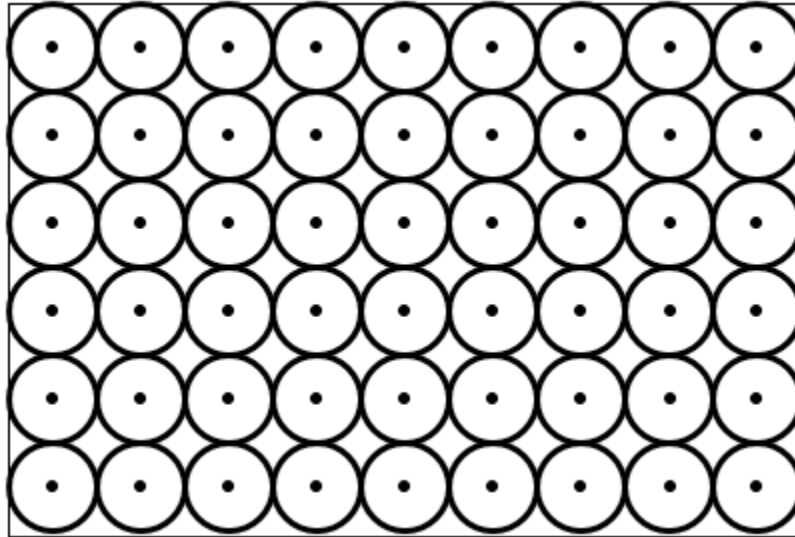


Figure 1.2 Full Coverage

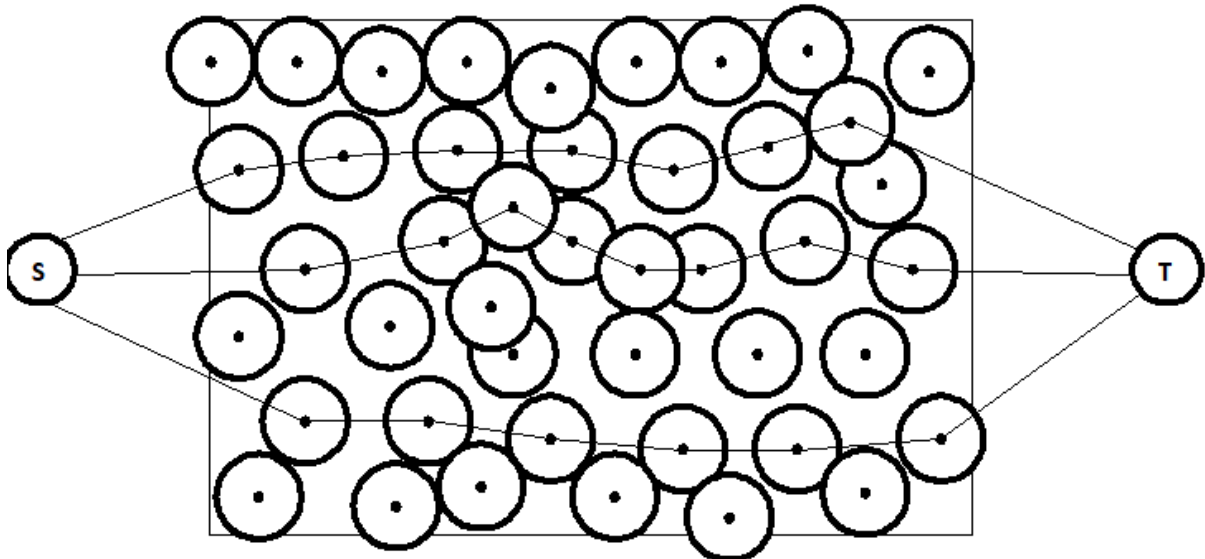


Figure 1.3 Barrier Coverage

Much fewer sensors required, to be deployed in the region in barrier coverage since the entire region coverage is not needed which is the major advantage of this system. A set of virtual nodes, which can be named as source and sink nodes, will be placed at left and right boundaries of the network respectively as shown in figure 1.3. Instead of left and right boundaries, these virtual sensors can also be placed at top and bottom or inner and outer of the network. To form a barrier from one end of the network to the other end, communication range of series of nodes should overlap. To achieve maximum coverage, key objective of barrier coverage should be to have almost every point of monitoring area under the sensing range of at least one sensor node.

1.3 Types of Barrier Coverage

Barrier coverage was first introduced in the context of robotics systems[1]. Although in [2], types of barrier coverage is sparse and is divided into two strong and weak barrier coverage, in this research five types of barrier coverage are mentioned which are

1. Weak Barrier Coverage
2. Strong Barrier Coverage
3. Local Barrier Coverage
4. Global Barrier Coverage
5. Crossed Barrier Coverage

1.3.1 Weak Barrier Coverage: In weak barrier coverage, there exists a coverage gap in the region provided and the event to be detected follows the direction opposite to that of barrier path. Therefore, weak barrier coverage detects events, which happens or travels in a straight opposite to the orientation of barrier path as shown in figure 1.4 (a). As mentioned in [3], this type of barrier coverage is not suitable for intrusion detection as an intruder may have the intelligence of knowing the orientation of barrier path in advance. Therefore, to avoid from being detected will follow the path where a coverage gap exist.

1.3.2 Strong Barrier Coverage: A strong barrier covered network is one in which there exist minimum one k -barrier path (minimum value of k is 1) from one end of the network to the other end. Therefore strong barrier coverage can detect event which happens or travels in arbitrary path irrespective of orientation of barrier. As a result of which the disadvantage of weak barrier coverage is eliminated and therefore strong barrier coverage is suitable for intrusion detection as shown in figure 1.4 (b).

1.3.3 Local Barrier Coverage: Given N sensor nodes randomly deployed over a rectangular region to achieve k -barrier coverage ($k \geq 1$), a barrier formed at one end of the region can neither guarantee that the network is completely covered by the barriers nor will have information of other barriers. Therefore a local barrier coverage although provides a strong barrier coverage but has a disadvantage that the coverage information of whole network is not provided.

1.3.4 Global Barrier Coverage: Opposite to local barrier coverage, a global barrier has the complete information of network coverage and therefore will have information of

number of barriers formed in the network after the initial random deployment of N nodes. Although global barrier coverage does not provide full coverage as mentioned in figure 1.2, but will guarantee that almost every event happening in the network can be detected. Also as mentioned in [4] every crossing path in the network can be detected by global barrier coverage irrespective of the level of intelligence an intruder.

1.3.5 Crossed Barrier Coverage: The notion of crossed barrier was introduced for the first time in [3]. As per authors, crossed barrier coverage is provided if and only if two barriers perpendicular to each other can be formed which will intersect exactly in the middle. Therefore, if there exists a barrier from left boundary to the right boundary of the network then one more barrier should exist from top of the network to the bottom as shown in figure 1.4(c). If there exists more than one crossed barrier path then it is equivalent to providing strong barrier coverage in the network.

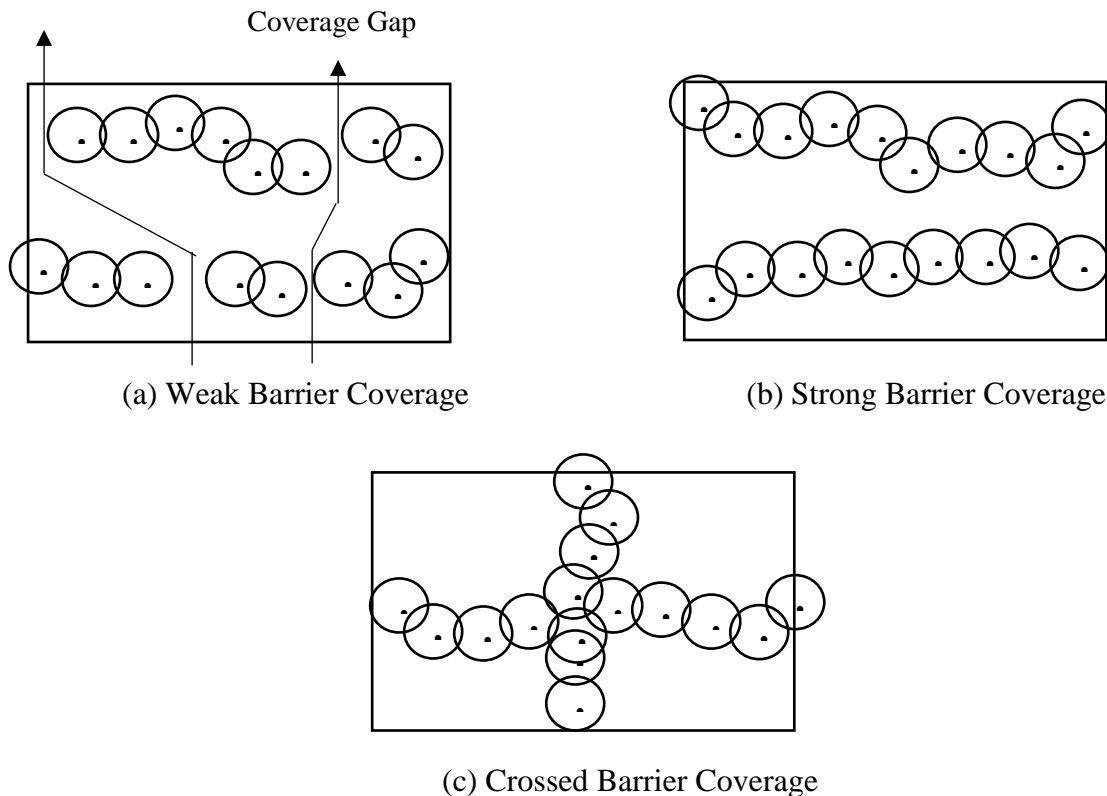


Figure 1.4. Types of Barrier Coverage

1.4 Techniques to achieve barrier coverage

There are a plethora number of techniques to implement barrier coverage in WSN. In this section, different techniques of barrier coverage are discussed. The available techniques can broadly be categorized as below:

1. Deployment Based Techniques
2. Sensor Based Techniques
3. Algorithm Based Techniques

These techniques are further divided as shown in figure 1.5.

In general, deployment techniques available for barrier coverage are deterministic and random. In deterministic deployment, coverage can be guaranteed only when number of sensors is less in number and area to be covered is less in size. In addition, the coverage area should be human accessible. In general, WSN are used in remote, human inaccessible and hostile environments in which random deployment is suitable. One such distributed deployment technique is proposed in [5]. Similarly Line Based Deployment Technique is proposed in [6], which is a special case of random deployment where sensors deployment is followed in a line from one end of the network to the other end of the network.

In barrier coverage, random deployment is indispensable. Hazardous context of barrier coverage do not allow sequential manual deployment of sensors, therefore random deployment is preferred over sequential deployment. In sequential or deterministic deployment the area to be monitored is always human-friendly. The sensor-based techniques discussed in literature are all post-deployment processing techniques, where first sensors are deployed and then processing in the form sensor movement or directional sensing is performed. For example in [7-12], authors have proposed the use of mobile sensors after deployment of static sensors in the network. Mobile sensors are used to strengthen the formation of barrier coverage. The problem associated with mobile sensor is the cost involved in terms of energy consumption in moving the sensor and finding the exact coordinates to deploy the moving sensor. In [13] a self-deployment approach of randomly deployed mobile sensors is proposed.

In this thesis, random deployment is used and a novel Minimum Non-Overlap Radius Deployment Algorithm (MNORDA) is proposed.

Directional Sensors, having a non-disk sensing area are also used in barrier formation. These sensors communication only in one particular direction opposite to that of omnidirectional communication which is a natural property of sensor. **In this thesis, omnidirectional communication is used.**

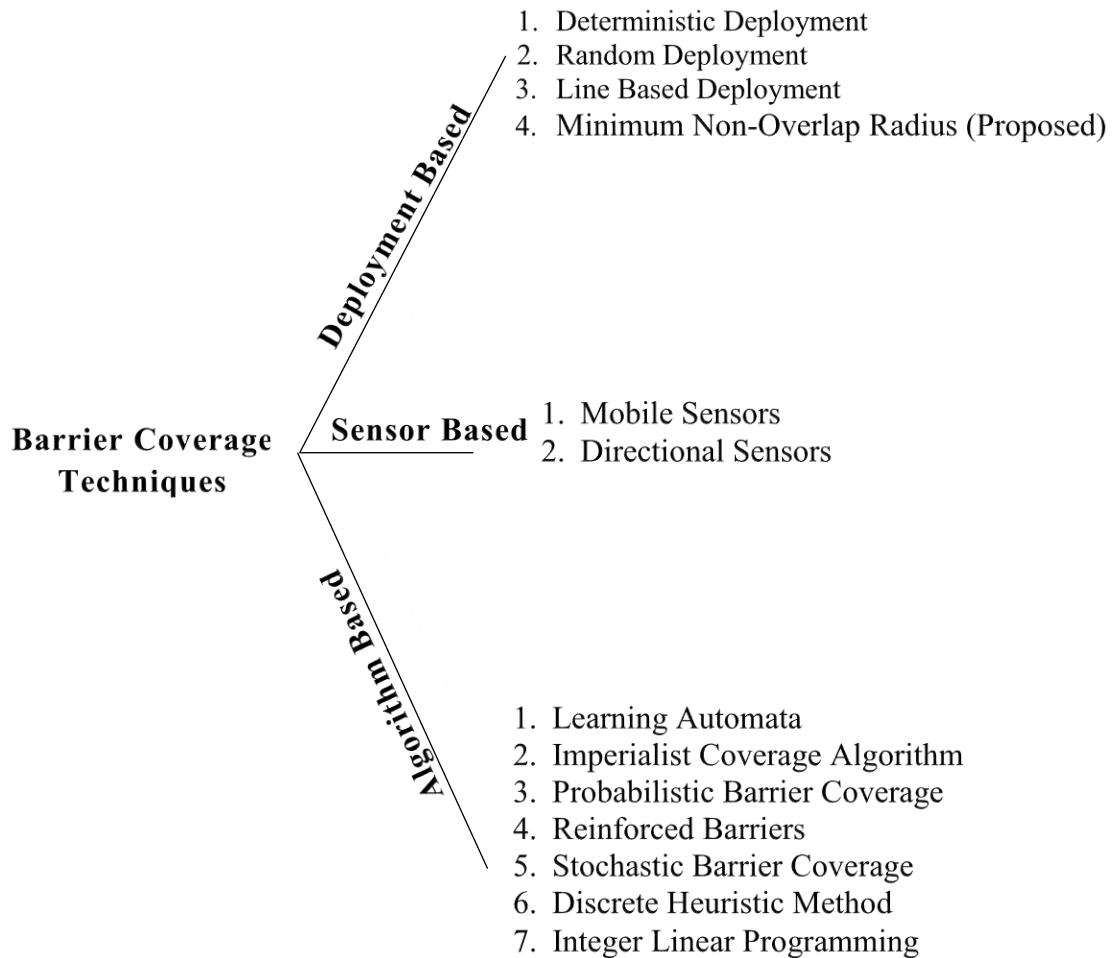


Figure 1.5 Techniques of Barrier Coverage

Algorithm based techniques also provides barrier coverage without the need of mobile and directional sensors in which algorithms can be developed for static as well as mobile sensors. The algorithms based on learning automata, imperialist coverage, stochastic coverage, integer programming are post-deployment techniques, which works on processing of communication data between sensor nodes.

1.5 Applications of Barrier Coverage

Applications of barrier coverage are enormous and it is not hard to foresee an exponential increase in the applications of barrier coverage. Although the list is huge, below mentioned are some important applications where barrier of sensor nodes can be formed and implemented in real time.

1. Critical Infrastructure Protection

2. Providing security in industry and homeland
3. Intrusion Detection in region of interest
4. Indoor Fire Detection
5. Forest Fire Detection and Monitoring
6. Chemical Leak Detection in Factories
7. Event and Movement Detection
8. Border Surveillance
9. Zone Monitoring in case of biological attacks
10. Animal Migration

1.6 Limitations of Barrier Coverage

Barrier coverage has attracted many researchers, as their applications are numerous. There are certain limitations associated with barrier formation as listed below

1. **Barrier Breach:** As mentioned in figure 1.4 (a), if the barrier path formation is known prior to its implementation then it is very easy to go undetected for any intruder. In these cases barrier coverage fails to provide security. Barrier breach problem is mentioned in [14, 15] and also their solution.
2. **Deployment Techniques:** As mentioned in section 1.4, barrier coverage supports random deployment and is indispensable. Therefore, in order to maximize the coverage and number of barrier paths to be formed, a good deployment technique is always necessary.
3. **Lifetime of Barriers:** Once the sensor nodes are deployed and involve themselves in forming barrier paths from one end of the network to the other end, the time for which continuous barrier coverage will be available has to be estimated. For this, the lifetime estimation of barrier path is required which needs computation of processing involved in forming and maintaining the barrier.
4. **Lifetime of Network:** The operating lifetime of network depends on the operating lifetime of individual nodes. In order to make the network lifetime last beyond the lifetime of an individual node, one of the prime requirements of barrier network is to operate and maintain the barriers paths formed for a long time. To exchange minimum data packets between nodes, energy harvesting techniques for each node can be implemented which can help in improving the lifetime of network

In this thesis, a technique that minimizes the data packets exchanged between nodes is proposed to enhance the overall operating lifetime of network.

5. Identification of Isolated Nodes: During random deployment of nodes, there is a possibility that one or more nodes become isolated and disconnected from the network. Identification of such nodes and permanently switching them off greatly reduces the power consumption and thereby increases the lifetime of network.

In this research, a novel technique, **probably** for the first time is introduced which will identify and detect the nodes that are isolated after initial random deployment.

1.7 Introduction to Hardware

In this thesis, hardware for barrier coverage is designed and developed for rural applications. Internet of Things (IoT) protocol named as Message Queueing Telemetry Transport (MQTT) is used to upload sensor data to IoT cloud to enable remote monitoring. MQTT is a two-way communication protocol, which facilitates in implementing remote controlling feature along with remote monitoring. The hardware is designed, implemented and tested for the following cases.

Case 1: When internet is present in the rural areas, then the barrier coverage can be formed to

1. Provide Wi-Fi range extension from the nearest place to the rural area through barrier of nodes
2. Forest Fire detection through IoT.

Case 2: When internet is not present

1. Forest Fire detection through multi-hop data transfer.

In the absence of internet, forest fire detection is implemented by the use of drones fitted with sensors and ZigBee (Xbee) transceivers. ZigBee is a Radio-Frequency (RF) device having a line-of-sight range between 90 meters to 1.6km. If a high gain antenna is interfaced then the line-of-sight distance can be increased to 64km also.

Therefore the focus of this research would be centered on the design and implementation of reliable, multifunctional, and yet power conservative network especially for rural areas. To achieve reliability, the network will be designed with sensors that will have nodes equipped with communication protocols i.e., ZigBee and Wi-Fi. Since Lifetime of any network is the

most important parameter, therefore this research focuses on design of energy management protocols for enhancing lifetime of the battery by minimizing the data exchange between sensor nodes. This research work will also emphasize on full coverage of the network by continuously providing k-barrier coverage where $k \geq 1$.

Chapter 2 gives the literature review of barrier coverage and the literature work of state-of-art algorithms is reviewed here. After summarizing the complete literature, work objectives are defined.

Chapter 3 defines and explains the methodology used to achieve the objectives. The working of proposed MNORDA algorithm is explained in detail along with example networks. Identification of isolated nodes and identification of shortest paths in terms of distance and in terms of number of nodes is also explained.

Chapter 4 gives the results of MNORDA algorithm where for different node size number of barrier formed are computed along with termination time, number of packets exchanged. Lifetime of each barrier is also calculated followed by analysis of MNORDA algorithm for different values of minimum non-overlap radius.

Chapter 5 is for implementing the hardware for rural application. Two cases of barrier formation with respect to hardware is discussed. One in presence of internet and one in its absence. When internet is present, a Wi-Fi range extender is designed followed by implementing concept of Internet of Things for forest fire detection. When internet is not present then forest fire is detected with the help of Zig-Bee.

Chapter 6 is for concluding the work of the thesis and then putting forward the scope of improvement in terms of future scope.

CHAPTER 2 - REVIEW OF LITERATURE

In this chapter, literature review on barrier coverage in Wireless Sensor Networks is presented. After defining the main preliminaries in the area of barrier coverage like, types of coverages, types of barrier coverages, techniques available for achieving barrier coverage in WSN and group wise classification of these techniques in chapter one, here in this chapter a detailed review of literature is presented.

In section 2.1 and 2.2 the techniques of barrier coverage based on centralized and distributed algorithms are reviewed followed by deployment based techniques in section 2.3, sensor based barrier coverage techniques in section 2.4 and algorithm based techniques in section 2.5.

2.1 Centralized Approach

In centralized approach, it is always necessary to assume that there is a central coordinating unit that has the ability to communicate with all the sensors in the region. Moreover, the central unit knows the position of all the sensors in the network.

In [6, 8, 16], a centralized algorithm first finds barrier gaps in the network while forming the barrier paths, from one end of the network to the other end. Once the gaps are found, mobile sensors are used to cover these gaps. Du, J., et al., in article [17] have proposed a novel k-discrete barrier coverage algorithm whose goal is cover specific points in the region. Identification of these discrete points is not shown and covering only specified discrete points does not guarantee the whole network coverage. One more centralized approach is presented in [18], where barrier coverage for linear domain is presented and the algorithm works for sensors with the same sensing range. Authors have focused on time complexity of the algorithms in their article. In centralized algorithm of [19], Li, S. and H. Shen have discussed barrier coverage problem for a 2-D plane suitable for border surveillance and here also the sensing range of all sensors are same. Time complexity of the proposed algorithm is also presented. As many algorithms available in literature are NP-hard the time complexity is an open question to define whether the algorithm designed is solvable in polynomial time or not. If an algorithm is not polynomial, solvable in time then it is an NP-hard problem. In this thesis, the proposed algorithm works for both uniform and non-uniform sensing range. In [20] the centralized algorithm proposed assumes that after random deployment, the initial position of sensors is known which is not close to practical scenario. Authors have also worked on energy-efficiency.

In [21] Nguyen, T.G., et al., focused on optimizing the number of barriers by proposing two algorithms, both being centralized. The first algorithm moves the sensors to cover the sensor gaps and the second algorithm forms the barrier paths. Bhattacharya, B., et al., of [22] have worked on proposing barrier coverage for intrusion detection on border area and have discussed an optimal method of moving the sensors to cover the gap created after deployment of sensors. The centralized approach presented [23], barrier formation for line based segment is discussed and authors have assumed that the destination is already known to the algorithm and also the radii of the sensors. Apart from barrier formation, work on minimizing the energy spent is shown in the article. In [24], He, S., et al., have proposed a novel curve-based deployment against the traditional line-based deployment for barrier coverage in WSN. Sensing model considered is heterogeneous where sensor range of sensors is not uniform. In centralized approach of [25], Dobrev, S., et al., have performed complexity analysis of barrier path formation using relocatable sensors. They prove that the sensor movement is NP-hard (Non-Deterministic Polynomial time-hardness).

In the centralized approaches discussed above, one problem is the assumption of a central node that has the information of all the deployed nodes that cannot be practical if the number of sensors are more. In addition, the performance analysis is centered towards the central node and therefore the performance of individual nodes is ignored. All centralized approaches have disadvantages such as high message overhead, single point of failure and lack of scalability. The work in this thesis is therefore centered towards distributed approach and sensors with different sensing ranges are used.

2.2 Distributed Approach

Distributed approach overcomes the disadvantages of centralized approach and therefore facilitates in providing the information of every node to every other node in the network. In distributed approach discussed in [26], Kong, L., et al. have worked on deploying mobile sensors by considering that number of sensors, are limited. Algorithm discussed in this article is fully distributed and based on virtual force and convex analysis. Line-based coverage, fully distributed algorithm is proposed in [27], where Jia, J., et al., prove that their algorithm reaches a stable deployment in finite time. Barrier formation is obtained by redistributing the mobile sensors among the stationary sensors. A distributed approach based on Integer Linear Programming model is used in [28] to solve the problem of k-barrier coverage with minimum

energy consumption. Ban, D., et al., here proves that the algorithm is an NP-hard problem and show how to relocate mobile sensors for barrier construction.

Two time-efficient distributed heuristic methods namely Line K-coverage MinMov and Line K-coverage MinMov+ algorithms which are proposed in [29] minimize the energy consumption during sensor movement. Line based 1-coverage problem is solved using these two distributed algorithms. Algorithm proposed in [30] coordinates the movement of sensor nodes to form barrier paths in the region with the available sensors. Algorithm proposed here tries to maximize the number of barrier paths in the region. An autonomous self-deployed sensors for barrier path formation is proposed in [31]. As per the assumption made in this algorithm, the barriers are self-formed after initial random deployment. In distributed approach presented in [32], He, S., et al. propose a scheduling algorithm where mobile sensor monitors each point on barrier coverage. Enhancement of barrier coverage is done by mobility of sensor node and arrival information of the intruder.

Barrier coverage for monitoring dynamic movements in real time is proposed in [33] which can be implemented in, say for example monitoring the marching troops. Cooperative movement of mobile sensors is proposed to maintain barrier coverage.

Dynamic sensor monitoring is proposed in [34] for maintaining barrier coverage when number of mobile sensors are sparse. The algorithm periodically monitors the barrier line with the help of mobile sensors. Directional Sensing strong barrier coverage algorithm with video cameras is proposed in [35] where the concept of virtual node for a 2-dimensional plane is introduced to construct a directional barrier graph. A novel push-pull-improve algorithm to improve the energy cost is proposed in [36]. The cost here is defined as energy consumption cost or any other cost associated with the sensor. Message complexity and time complexity of the proposed algorithm has been calculated. A fully distributed approach to enhance the energy efficiency in barrier coverage is proposed in [37] where the prime objective of the research is to increase the lifetime of barriers. In distributed approaches use of mobile sensor is dominated which needs additional overhead and moving cost of sensor. This although guarantees the barrier formation but hampers the lifetime of network.

The MNORDA algorithm proposed in this thesis is purely a distributed algorithm where every node communicates only with its neighbors. The sensor nodes used are all stationary.

The philosophy behind distributed and centralized approach is the presence and absence of coordinating node. In centralized approach, a coordinating node capable of communicating with all other nodes is present whereas in distributed approach it is absent.

2.3 Deployment Based Approach

Random deployment has always been preferred in barrier coverage for the obvious reasons mentioned in section 1.4. Certain special cases of random deployment available in the literature are

1. Line-Based deployment
2. Curve-based deployment
3. Autonomous Deployment
4. Self-Deployment

2.3.1 Line-Based Deployment: Line based deployment proposed in [6, 8, 17-19, 27, 29, 38] follows a basic rule that all the sensors should be deployed in a line from one end of the network to other end say for example from an aircraft. During such a deployment, the distribution of sensors will mostly be non-uniform and sensors will be deployed very close to each other, which creates redundancy. To remove such redundancy mobile sensors are required. Also maximizing barrier coverage in line-based deployment is not possible. Extra efforts are required during the deployment process to find the shortest line between two ends of the network.

2.3.2 Curve-Based Deployment: Curve based deployment in barrier coverage is considered as a special case of line based deployment. It was proposed for the first time by the authors of [24]. He, S., et al., proves that line based deployment is sub optimal and provides the proof that curve based deployment is better than line based. Although curve based deployment is better than line based but it still does not guarantee maximum barrier coverage in the region of interest therefore it is not considered as an optimal solution in providing maximum barrier coverage.

Note: Proposed MNORDA algorithm is considered as optimal as it provides maximum number of barriers when compared to all other techniques discussed in the literature. Optimality of MNORDA is proved by forming maximum number of barriers while achieving maximum lifetime of each barrier.

2.3.3 Autonomous Deployment: An efficient autonomous deployment algorithm is proposed by the authors in research article [39] for homogenous network, and by Bartolini, N., et al., of [40] for heterogeneous network. A homogenous network is one where the range of each sensor is same and opposite is heterogeneous network. In the proposed algorithm of [40], each sensor by itself will make movement decisions during

deployment without any need of prior knowledge conditions of operating. The algorithm works well when sufficient number of sensors are available for barrier formation and terminates in finite time. Mobibar is another algorithm for autonomous deployment proposed in [30]. This algorithm helps in deploying the sensor nodes randomly in the region of interest and helps in increasing the number of barriers formed. One drawback associated with these algorithms is that they do not guarantee maintenance of minimum distance between two nodes as proposed in MNORDA. This minimum distance helps in spreading the sensor nodes uniformly and evenly throughout the region of interest.

2.3.4 Self - Deployment: In self-deployment technique mobile sensors are made to self-coordinate to achieve barrier coverage. It is more commonly known as consensus technique where each node follows a local information. One such technique is discussed in letter [41]. Rout, M. and R. Roy, in [13] proposed two schemes of self-deployment based on binary sensing model for barrier coverage. Scheme one is virtual force where a sensor moves in the direction of virtual forces along the X and Y direction of 2-D plane. Maximum distance that a sensor can move is fixed. Scheme two is based on potential theory which has three phases namely i) Computation of direction of movement ii) Computation of distance to be covered iii) Sensor movement.

The philosophy behind the deployment algorithms is to deploy the sensor nodes in such a way that at least one barrier can be formed from one end of the network to the other end.

2.4 Sensor Based Approach

This section covers barrier coverage techniques that can be achieved by using Mobile Sensors and Directional Sensors. In mobile sensor based approach, the sensors will be moved either after initial random deployment or during deployment phase. An extra effort in terms of cost associated with energy is needed to move sensors during or after deployment. Second technique is by using directional sensor, where every sensor will sense and communicate only in one particular direction. Therefore, barrier path formation strongly depends on the directional sensing property because of which the number of barriers formed will always be less than an omnidirectional sensor. The proposed MNORDA algorithm uses an omnidirectional sensor and stationary sensor thus is able to achieve highest number of barriers along with longest operating lifetime of barriers in terms of weeks, when compared to other state of art works proposed in the literature.

2.4.1 Mobile Sensor Based Approach: In the research work presented in [42], Chang, C.-Y., C.-Y. Hsiao, and Y.-T. Chin., have proposed formation of defense barriers for intrusion detection using mobile sensors. All the sensors move in a distributed manner at the initial phase of algorithm to form k-barriers. The proposed work aims at enhancing the lifetime of barrier while forming k-barrier paths. Ma, H., et al., of [43] proposed two heuristics, one using stationary sensor to study minimum energy cost problem for forming k-barrier coverage, and forming maximum k-barrier paths using limited mobile sensors. Both the heuristics follow Integer Linear Programming (ILP). Saipulla, A., et al., of [6, 8] proposed barrier coverage technique using sensors with limited mobility. Draw of the proposed approach is that sensors movement is line based. In the work proposed in [28], sensor movement is carried for relocation work to achieve energy efficiency in the network and energy efficiency is devised by constructing one barrier coverage. The work addressed in article [44] is focused on prolonging the lifetime of barrier coverage by using energy-plentiful mobile sensors and energy-scarce stationary ground sensors. The work has proposed two algorithms that are compared with each other for performance evaluation. Redistribution of sensors after deployment is studied in [27], where Jia, J., et al., first work on finding an optimal layout for deployment for the given random deployment scenario. Then, line-based barrier coverage is achieved by redistributing the sensors. Wang, Z., et al. in [45], finds the errors in location after initial deployment and will move mobile sensors for barrier formation if and only if the network is not barrier covered.

In the research work presented in [17], Du, J., et al., have proposed a novel k-discrete barrier coverage algorithm whose goal is to cover specific points in the region by using information of redundant sensors. In this work, mobile sensors are used to cover some specific points by deploying them in k lines. Identification of these discrete points is not shown and covering only specified discrete points does not guarantee the whole network coverage. In [18], Chen, D.Z., et al., have proposed algorithm to minimize maximum sensor movement but this algorithm works only for line based barrier coverage. Li, S. and H. Shen. of [19] tries to minimize the maximum sensor movement to save power consumption of the sensor and achieve barrier coverage in 2-D plane. The work proposed in [5] focused on relocating the sensor after initial random deployment. A novel clustering technique is proposed whose results are promising. Two time-efficient distributed heuristic methods namely Line K-coverage MinMov and Line K-coverage MinMov+ algorithms are proposed in [29] to minimize the energy

consumption during sensor movement but works only for Line based 1-coverage problem.

2.4.2 Directional Sensor Based Approach: Han, R., L. Zhang, and W. Yang. in article [46] proposed Weighted Barrier Graph (WBG) approach to use camera as directional sensors for barrier coverage. Two scenarios have been studied, one when only stationary sensors are deployed, the number of mobile directional sensors required for k-barrier coverage and second when both stationary and mobile sensors are deployed, maximum number of barriers formed. Tao, D., et al., in [35] have used video cameras as directional sensors to form barrier coverage, where the authors find the appropriate orientation of directional sensors to form strong barrier coverage. Zhang, L., et al., in [47] and Han, R., L. et al., in [48] proposes a strong barrier coverage network to maximize k-barrier coverage and enhance the operating lifetime of barriers.

In sensor-based approach, the philosophy followed is the use of mobile and directional sensors for barrier formation.

2.5 Algorithm Based Approach

In this type of approach for forming barrier coverage various algorithms have been proposed which might work for mobile sensors, stationary sensors and for directional sensors. The algorithms used by the researchers for formation of k-barriers and enhancing the lifetime of these barriers are listed below.

1. Learning Automata
2. Imperialist Competitive Algorithm
3. Probabilistic Barrier Coverage
4. Reinforced Barriers
5. Stochastic Barrier Coverage
6. Discrete Heuristic Method
7. Integer Linear Programming

2.5.1 Learning Automata Approach: Learning automata is a stochastic based machine learning approach that works on adaptive decision making to enhance performance of possible actions. One such technique to enhance the number of barrier paths is discussed in [37, 49] and applying the same technique to enhance lifetime of barrier

coverage by implementing sleep scheduling is discussed in [50]. Learning automata can also be used to monitor the coverage provided in wireless sensor network as discussed in [51]. Maximizing the lifetime of target coverage is also possible by learning automata approach as shown in [52]. This technique has also been used in deployment technique to improve k-barrier coverage in WSN as discussed in [53].

2.5.2 Imperialist Competitive Algorithm: Imperialist Competitive Algorithm is considered as a special counter part of Genetic Algorithm and therefore is very much suitable for barrier coverage applications. Genetic Algorithm is based on biological evolution and Imperialist algorithm is based on human evolution and needs a mathematical model and computer simulation. Mostafaei, H., et al., in [54] have shown how to use imperialist algorithm not only to implement barrier coverage but also to enhance the number of barriers formed. Simulation results obtained by the authors are promising.

2.5.3 Probabilistic Barrier Coverage: In this type of technique, detection probability is calculated which gives the information for intrusion detection through barrier formation. One such technique is proposed in [55], where minimum weight barrier algorithm is designed to solve the problem of barrier coverage scheduling which is otherwise a NP-hard problem. A protocol to provide global barrier coverage is also proposed. A k-barrier coverage protocol to enhance the energy efficiency along with maintaining the coverage is shown in [56]. Sheu, J.-P. and H.-F. Lin. here define a confidence probability, which is the minimum probability of providing the coverage with at least k-sensors.

2.5.4 Reinforced Barriers: Reinforced barriers are new type of barriers as mentioned in [57]. In this work, Kim, H., J.A. Cobb, and J. Ben-Othman, first define that an intruder can take any arbitrary movement in the network and therefore the traditional barrier coverage cannot detect it. To detect the movement variation of intruder, reinforced barriers are created and inserted in the network which detects the arbitrary movements the intruder can take to cross the network. Authors of this article also propose four different approaches to construct reinforced barrier for the given layout of sensor and then compare their relative performance.

2.5.5 Stochastic Barrier Coverage: Stochastic barrier formation is related to study of having a random probability distribution to be analyzed statistically. A statistical analysis is applied to patterns where precise prediction is not possible. Mostafaei, H., in [37] proposed stochastic based barrier coverage by using distributed learning automata. The prime objective of the study was to enhance the energy efficiency of the barrier

network. Simulation results of the work show that the proposed algorithm outperforms the optimal method.

2.5.6 Discrete Heuristic Method: This type of heuristic method was first proposed in [7] where dynamic coverage algorithm based on Discrete Heuristic Method is proposed. In this technique, first, a set of fixed sensors will be deployed along the line and then mobile inspection sensors are made to walk to detect any barrier gaps. This proposed algorithm was especially for monitoring network coverage in transmission line of smart grids.

2.5.7 Integer Linear Programming: In linear programming technique, objective is find solutions to problems using linear functions, say for example a line. Integer linear programming is one where all the variables are strictly integers. When all the variables are, binary i.e., 0 or 1 then it is called as 0-1 Integer Linear Programming or Binary Integer Linear Programming. One such 0-1 Integer Linear Programming is used in [58] where authors use the technique to achieve k-barrier coverage in WSN. Authors claim they are first to find a limit on number of sensors required to form barrier coverage. In the article [43] the problem of maximizing and solving k-barrier coverage is studied and solved as Integer Linear Programming. Ban, D., et al., in article [28] use linear programming to construct k-barriers with minimum energy consumption.

Table 2.1 Comparison of algorithm based approaches

S. No	Algorithm	Advantages	Disadvantages
1	Learning Automata	Technique can be used to enhance number of barriers as well as lifetime of barrier coverage	In literature the technique is used for target coverage and
2	Imperialist Competitive	Technique can be used to enhance number of barriers as well as lifetime of barrier coverage	The algorithm is a counter part of very old genetic algorithm and the advancements of it is missing in literature.
3	Probabilistic Barrier	This technique is used for solving NP-hard barrier coverage scheduling problem	Used for intrusion detection only.
4	Reinforced Barrier	Any arbitrary path taken by the intruder can be detected by creating reinforced barriers	Used for intrusion detection only
5	Stochastic Barrier	Distributed learning automata technique forms stochastic barrier coverage	Precision prediction is not possible

6	Discrete Heuristic	Discrete heuristic method solves the problem of dynamic coverage	Mobile sensors are mode to walk to detect barrier gaps which needs high energy efficiency
7	Integer Linear Programming(ILP)	Using ILP a limit on number of sensors required can be found	ILP a NP-Hard problem

The algorithm-based approaches follow a philosophy that the proposed techniques can be used for mobile sensors, directional sensors or sensors that are not movable. The idea is to propose technique in-terms of algorithms that are suitable for barrier formation in any of the above-mentioned case.

After putting forward the detailed review of all the required topics of barrier coverage, viz., deployment schemes, techniques of barrier coverage, algorithms to maximize k-barrier coverage, enhancing the lifetime of barriers, a very brief review of other techniques available in the literature which may be useful for barrier coverage is presented. The techniques are

1. **Ant Colony Optimization:** Ant Colony Optimization (ACO) discussed in article [59] is a probabilistic optimization technique based on graph theory. As ants and bees belong to colonies which are self-organized, authors of this article use ACO for node deployment technique which is suitable in formation of barrier coverage.
2. **Artificial Fish Swarm:** It is another self-organized system which is decentralized and is used in [60] for coverage optimization. 2. Artificial Fish Swarm technique is based on swarm intelligence and ant colony is an example of swarm intelligence.
3. **Hybrid Particle Swarm Optimization:** 3. Hybrid Particle Swarm Optimization is a technique, which is more suitable to be used with mobile sensors as in this each movement of sensor, is influenced by local information. This information helps sensors to find a best possible position in order to form barrier coverage. Maleki, I., et al., in article [61] have used hybrid particle swarm optimization for solving the coverage problem.

2.6 Summary of Literature Review

In this section, a brief comparison of research works, which are very close to the work presented in the thesis, is summarized. The result of the proposed algorithm is compared with the state-of-art works as briefed below.

1. Distributed Deployment Algorithm for Barrier Coverage (DDABC) given in [5] proposed a novel clustering technique where sensor nodes are assigned evenly to each cluster. The work proposed here is a pre-deployment process that requires sensor movement after initial random deployment. The main goal of the work is to relocate

the sensor from its initial position into the clusters with minimum relocation cost and with an aim to maximize sensor barriers. DDABC consists of five phases starting from cluster formation phase, packet exchange phase, cluster and cell adjustment phase and finally the sensor-moving phase.

2. A Centralized Barrier Coverage Algorithm based on Distributed Learning Automata (CBCDLA), is proposed in [49] which works on adjusting the orientation of directional sensors, such as cameras, in a non-overlapping form to form a strong barrier coverage. The distributed version of the algorithm is also presented. The results of centralized version of the algorithm are promising as they outperform the previously proposed algorithms. The algorithm proposed here assumes that nodes are already deployed randomly in the entire network and therefore works only on maximising the number of barriers formed.
3. An autonomous self-deployment algorithm named as MobiBar, with stationary and mobile sensors is presented in [30]. In this algorithm, coordination among stationary and mobile sensor is used to achieve self-deployment of sensors nodes to achieve maximum barrier coverage in the network. Authors of this article have also proposed MobiBar algorithm terminates in finite time. MobiBar has the ability to self-configure and self-heal in cases of sensor failures.
4. Imperialist Coverage Algorithm for Barrier Coverage (ICABC) proposed in [54] uses sensor node for monitoring the barrier coverage. The main objective of the algorithm is to enhance the operating lifetime of overall network. Therefore the algorithm proposed in this article assumes that the nodes are already deployed and barrier are already existing. The authors here have assumed that one barrier will last for one week, therefore lifetime of network will be number of barriers multiplied by number of weeks. Overall time complexity of imperialist algorithm is also presented.
5. Ban, D., et al., in article [28] have proposed Approximate to Horizontal and Vertical Grid Barrier (AHVGB) in which emphasis was given to construct horizontal and vertical grid barriers independently. In this algorithm gaps exist between two horizontal barriers therefore additional mobile sensors are required to cover these gaps. The network here is first divided into sub-regions and barriers constructed in each sub-region are all local barriers without any communication with other sub-regions. At first the authors present an energy efficient 1-barrier coverage algorithm and then extend it and present a divide and conquer algorithm for achieving energy efficient k-barrier coverage.

6. Maximizing the barrier coverage using directional sensor network is shown in [48] where efficient algorithm using two-round maximum-flow algorithm (TMFA) is presented. The directional sensors used are rotatable with non uniform lifetimes. As per the authors of this article sensors in practical scenario has been considered where lifetime of each sensor is different. TMFA algorithm finds the maximum barrier number while maximizing their operating lifetime.
7. Achieving barrier coverage using line-based sensor deployment has been studied in [6] where sensors are deployed in a line through an aircraft. The algorithm proposed by the authors first will deploy sensors in a line and then will find the coverage gaps so that mobile sensors can be moved to cover these gaps to improve barrier coverage. While forming barriers, the algorithm tries to balance the energy consumption among the mobile sensors. The coverage provided by the algorithm follows a straight line from one end of the network to the other end and therefore cannot guarantee full coverage of the network. The proposed algorithm has been named as Mini-max (Minimax) algorithm for ease of referencing.
8. Mobile sensor to form barrier coverage automatically has been used in [26] where Kong, L., et al. propose a virtual-force based algorithm (VF). The main objective of the algorithm is use limited number of available mobile sensor to form barrier coverage with highest detection probability. The proposed algorithm works on cooperative scheme to relocate the mobile sensors from their initial random positions so as to uniformly deploy the sensors throughout the network. The work proposed in this article has focused on deployment scheme to form barrier coverage in the network with mobile sensors.
9. Disjoint-Path (Dpath) algorithm proposed in [47] works on forming a coverage graph for directional sensors to model barrier coverage. Both centralized and distributed algorithms are presented to solve the problem of barrier coverage using integer linear programming. Authors of this article focused on maximizing the number of barrier in the network and the simulation results shown are promising. The simulation results obtained proves that Dpath algorithm provide close-to-optimal solution to barrier coverage problem.

Table 2.2 gives the summary of literature review concisely.

Table 2.2: Summary of Literature Review

Ref.	Author	Year	Issue Addressed	Technique used	Result Compared
[5]	<i>Nguyen et. Al.</i>	2018	Distributed Deployment for barrier coverage	Mobile Sensors	Number of barriers formed
[6]	<i>Saipulla et. Al.</i>	2010	Maximizing number of barrier paths, lifetime and reducing convergence time	Line based sensor deployment	Energy Consumed and Termination Time
[26]	<i>Kong L et. Al.</i>	2009	Reducing convergence time	Mobile Sensors	Energy Consumed and Termination Time
[28]	<i>Ban D et. Al.</i>	2010	Maximizing lifetime of barrier paths	Mobile Sensors	Number of barriers formed, Energy Consumed and Termination Time
[30]	<i>Silvestri et. Al.</i>	2017	Maximizing number of barrier paths, lifetime and reducing convergence time	Mobile Sensors	Number of barriers formed, Energy Consumed and Termination Time
[47]	<i>Zhang et. Al.</i>	2009	Strong Barrier Coverage	Directional Sensing	Lifetime of Barriers
[48]	<i>Han et. Al.</i>	2016	Maximizing number of barrier paths	Directional Sensing	Lifetime of Barriers
[49]	<i>Khanjary et. Al.</i>	2018	Maximizing number of barrier paths formed	Learning Automata	Number of barriers formed
[54]	<i>Mostafaei et. Al.</i>	2017	Maximizing lifetime of barrier paths	Imperialist algorithm	Lifetime of Barriers

2.7 Conclusion of Literature Review

In this section the gaps associated with the works discussed in the literature is put forward. The techniques discussed in literature are all post-deployment techniques, which needs additional amount of work after deployment to form barrier coverage. This additional work may be in the form of utilizing mobile sensors, using directional sensing property of sensors with cameras, etc.

In line-based deployment techniques, sensor nodes are deployed only in a line and there is no guarantee whether all the sensors are really deployed in a line or not, as it is a random deployment, say from an aircraft. Also only limited number of sensors can be deployed in a line. One more problem is that there is every chance that nodes are deployed very close to each other which will result in sensor redundancy.

Utilizing mobile sensor will put an extra effort for moving the sensor to the desired location. The extra effort will be in terms of energy cost, extra data packet cost and movement cost. In addition, to move a mobile sensor first the barrier gap has to be identified and after which the mobile sensor has to be moved exactly to the same place where barrier gap is present, a task which is never 100% assured.

Directional sensing property of sensor needs cameras and special antennas which can sense the information in the particular direction. Barriers formed with directional sensing will always be less in number for the given number of sensors. In addition, to operate the sensors with directional property extra power source is required, which will directly hamper the operating lifetime of overall network.

Learning Automata and Imperialist Algorithm although looks very promising, testing of these algorithms in real time, except simulation, has never been reported.

2.8 Objectives

Based on the literature review, its summary and conclusion, the objectives of this research work are

1. To distribute nodes evenly throughout the region of interest for random deployment during pre-deployment phase.
2. To construct k-barriers paths for multi-hop data transfer from source to sink ($k \geq 1$)
3. To enhance operating lifetime of barrier paths.
4. To provide information of every node to every other node in the network by forming adjacency matrix of the network.
5. To identify isolated nodes after initial random deployment.
6. To identify shortest path in terms of number of nodes and distance.

CHAPTER 3 - METHODOLOGY AND ALGORITHM DESCRIPTION

This chapter explains the methodology adopted for achieving the objectives along with the description of algorithm. The proposed, novel Minimum Non-Overlap Radius Deployment Algorithm (MNORDA) is explained in detail in this chapter.

3.1 Methodology

To achieve the objectives defined, the methodology followed is

1. To achieve a uniform deployment of nodes throughout the region of network with certain predefined constraints.
2. To form k-barriers, two virtual nodes, named as source node s and sink node t is deployed on the left boundary and right boundary of the network respectively. Then starting from source node s traversing to sink t is implemented through the intermediate nodes in the network.
3. To enhance the operating lifetime of network, lifetime of barrier paths are enhanced which will prolong the overall operating lifetime of the network. For this a novel techniques is implemented where a node will share data packets consisting information of network, with neighboring nodes if and only if, the neighbor node did not receive the information, say from some other neighbor. This will help in avoiding duplicate data packets, which will anyhow be discarded by the nodes. As a result, there will only be data packets in the network without any overhead or duplicate packets, thereby increasing the lifetime of network.
4. To generate adjacency matrix, every node will first generate its own row which will have information of its neighbors. Then with the help of step 3 defined above adjacency matrix will be generated.
5. In the adjacency matrix if any row and its column has all 0 entries then the corresponding node will be treated as an isolated node. The adjacency matrix of an isolated node will have all 0's.
6. Shortest path is identified in terms of distance as well as in terms of number of hops, which is number of nodes for a k-barrier path from source to sink node. Shortest path

in terms of distance is calculated based on the difference between coordinates of successive nodes in that path.

3.2 Functions in MNORDA algorithm

In this section, a detailed, explanation functions which are called during the different phases of the proposed MNORDA algorithm is illustrated.

3.2.1 Function inRange: This function is executed to find the nodes that are in range for which following steps are required.

1. Consider the node set $S = \{s_1, s_2, \dots, s_n\}$ as N nodes with range R deployed in area with Length L and width W . The proposed algorithm works for both rectangular area (when $L \neq W$) and for square area (when $L = W$).
2. Consider edge set $P = \{P_1, P_2, \dots, P_n\}$ as the set of barrier paths that can be formed between nodes if their range overlaps.
3. Consider two virtual nodes \hat{s} and t at the left and right boundaries respectively of area under consideration. Node \hat{s} is called as source node with a range R_s whose value is at least five times than the range of individual nodes. Node t is sink node with range R_t equal to that of source node. Sink node can also be termed as destination node. Range of source node and sink node are kept high so that they can communicate with maximum number of nodes on the left and right boundaries of the network.

The inRange function checks whether two nodes are in range by checking the difference between their coordinates and then comparing it with the range R . Consider Eq. 3.1 and 3.2 given below:

$$\begin{aligned} abs(S_m.x - S_n.x) &\geq S_m.range \quad \forall V \\ or & \dots(3.1) \end{aligned}$$

$$abs(S_m.x - S_n.x) \geq S_n.range \quad \forall V$$

$$\begin{aligned} abs(S_m.y - S_n.y) &\geq S_m.range \quad \forall V \\ or & \dots(3.2) \end{aligned}$$

$$abs(S_m.y - S_n.y) \geq S_n.range \quad \forall V$$

Where x and y are the coordinates of node S_m and S_n .

S_m, S_n are nodes from node set.

S_m .range, S_n .range are range of node m and n respectively

V is total number of nodes in the network.

Note 1: In Eq. 3.1 and 3.2 range of both the nodes, S_m and S_n is considered to prove that this algorithm works for network with homogenous nodes and network with heterogeneous nodes.

Note 2: Absolute value of the difference in coordinates is considered to prove that in this algorithm all the nodes are having Omni-Directional communication.

3.2.2 Function sendHello: Function inRange returns a true value if the nodes are in range and false otherwise. If a true value is returned for two nodes S_m and S_n , then this function will send a hello packet from node S_m to node S_n . If a response to this hello packet is received by node S_m then it means that nodes S_m and S_n are in range. Accordingly the node S_m will update its list of neighbors.

3.2.3 Function Graphics: This function helps in setting the graphics for simulation environment. There are multiple parameters used in the simulation environment where the algorithm is implemented. Definitions of such parameters is given below

1. Dia: Diameter of the circles, which represent range of nodes in the simulation.
2. Sdia: Diameter of the source and sink nodes as their range is larger than normal nodes.
3. X_{offset}: Variable to set offset for x coordinates so that the nodes aren't drawn too close to the edges of the region (in the simulation environment, the screen used to display graphics is considered as region).
4. Y_{offset}: Variable to set offset for y coordinates so that the nodes aren't drawn too close to the edges of the screen.
5. X_{max}: maximum x coordinate of the region equivalent to Length L .
6. Y_{max}: maximum y coordinate of the region equivalent to Width W .
7. In figure 3.1 (a) and (b) Sdia is far greater than range value.
8. Node Id: Numeric Id of each node which is inserted in the middle of node circle. The size of node id is dynamic and can be increased or decreased depending on requirements of graphics to be displayed in the result output.

9. Pathtype: Can be 0 or 1 for edge-disjoint and node-disjoint paths respectively in the simulation.



Figure 3.1 Nodes with their range circles

Figure 3.1 (a) represents a sensor node randomly deployed in the network. The inner circle represents the node and the numeric digit inside the circle is the node id. The outer circle is the range of that node.

Figure 3.1 (b) represents the virtual node with a green inner circle. Node ids of virtual source and sink nodes are always kept as 0 and 1 respectively. The range of virtual nodes is very high when compared to normal node for the obvious reason as mentioned in section 3.2.1.

3.2.4 Function Layers: The proposed MNORDA algorithm is very versatile and offers the implementation of various functionalities layer wise in the graphics of simulation. For example,

1. Node_layer: This layer is to enable or disable drawing of the individual nodes on the screen.
2. Mesh_layer: Mesh layer is to enable or disable drawing of paths between nodes within range.
3. Path_layer: Path layer is to enable or disable drawing of paths between source and sink nodes.
4. Range_layer: This layer is to enable or disable drawing of range circles around nodes.
5. Label_layer: This layer is to enable or disable drawing of node ID labels inside nodes.
6. Adjacency Matrix_layer: This layer is to draw the adjacency matrices of nodes in the simulation graphics. For this, the number of nodes should be less, probably less than 15, if a 15 inch laptop screen is used for the simulation. As

the number of nodes increases the graphics goes out of the screen, as each adjacency matrix will of order $V \times V$, Where V is number of nodes. As there is, a minimum radius to be maintained between two nodes, as number of nodes increases the algorithm may not be able to deploy the nodes within the screen size and therefore will cease to run.

3.2.5 Function Node Parameters: This function provides the node ids to the node which is represented inside the node circle as shown in figure 3.1 (a) and (b). Also this function provides the node range and other important parameters as shown below

1. **Vnodedistx:** This parameter is to set the offset of virtual nodes from other nodes in the network. Virtual nodes are placed outside and away from the main network of nodes.
2. **Vnoderange:** This parameter sets the range of virtual nodes. Virtual nodes have increased range so that they can communicate with multiple nodes on the edge of the scattering environment.
3. **No-overlap:** This is a very important parameter of the algorithm and it is used to enable or disable scattering of nodes too close to one another. If enable then no two nodes will be deployed very close to each other.
4. **Overlapradius:** If No-overlap is set to true, this parameter defines the minimum deployment distance between the nodes
5. **Scattering:** To enable or disable random scattering of nodes on the simulation arena. For random deployment, scattering parameter should be enabled and for sequential deployment it should be disabled.
6. **Label Nodes:** This parameter provides unique ids to each node. The node ids will start from 2 as 0 and 1 id is given source and sink node respectively.

3.3 Algorithm Description

In the proposed Minimum Non-Overlap Radius Algorithm (MNORDA), there are eight different phases. Each phase can be considered as one algorithm. The eight phases of MNORDA algorithm are

1. Node Position Set
2. Find Neighbor Algorithm

3. Tell-Neighbor Algorithm
4. Generate Adjacency Matrix
5. Implement Symmetricity in adjacency matrix
6. Construct Barrier Paths
7. Identify isolated nodes in the network
8. Identify Shortest Paths in terms of distance and number of nodes

3.4 Node Position Set

In random deployment of nodes, the coordinates that each node takes are also be random. In this random deployment there certain problems which needs to be addressed, which are

1. A possibility that a node may be deployed outside the network or region of interest.
2. A possibility that a node may be deployed on the edge of the network or region of interest.
3. A possibility that a node may be deployed close to the edge of the network or region of interest.
4. Two nodes may take the same coordinates or may be deployed very close to each other

To avoid these problems certain constraints are implemented in the algorithms which are very useful in increasing the number of barriers formed. In the next section, first the constraints are defined and then a proof is given that if the constraints are not followed then there are coverage gaps or coverage holes in the network and also certain nodes gets deployed outside the region plus there are significant number of isolated nodes. By implementing the constraints, it will be proved that the coverage of the network is increased by removing the coverages holes created and there is significant reduction in number of isolated nodes.

The constraints implemented are defined as follows

1. **Constraint 1:** The distance between coordinates of two nodes should be greater than or equal to minimum non-overlap radius (O_R). O_R can be considered as minimum distance to be maintained between two nodes. This constraint will guarantee that no two will take the same coordinates and at the same time will not deployed close to each other.

$$\sum_{n=0}^V \sqrt{(j_{i+1} - j_i)^2 + (i_{i+1} - i_i)^2} \geq O_R \quad \forall n \quad \dots (3.3)$$

Where:

O_R – Minimum non-overlap Radius.

V – is the number of nodes.

i, j – are the coordinates of current node.

$i+1, j+1$ – are the coordinates of neighbor node

As per Eq. 3.3, every node will be assigned coordinates whose difference from its neighboring node coordinate should be at least greater than or equal to Minimum-Overlap Radius O_R .

- 1. Constraint 2:** Nodes should not be deployed close to the edges, on the edges and beyond the edge of the region of interest.

$$\left. \begin{array}{l} i < X_{offset} \\ j < Y_{offset} \end{array} \right\} \forall V \quad \dots (3.4)$$

$$\left. \begin{array}{l} i < X_{max} \\ j < Y_{max} \end{array} \right\} \forall V \quad \dots (3.5)$$

Where:

X_{offset} - maximum allowed distance (in terms of pixels on simulation screen) from the edge of region in horizontal direction.

Y_{offset} - maximum allowed distance (in terms of pixels on simulation screen) from the edge of region in vertical direction.

X_{max} - Length of the region L.

Y_{max} - Width of the region W.

As per Eq. 3.4 and 3.5, no node will be deployed close to the edge of the region or on the region; similarly no node will be deployed beyond the region of interest. All the nodes will lie well within the limits of the network.

Example 1: To prove the effectiveness of the constraints defines, consider a practical network with 20 nodes, with node ids 0 and 1 given to source and sink nodes respectively and node ids 2 to 19 given to all other nodes.

Figure 3.2 represents the random deployment of 20 nodes without implementing the constraint 1 and constraint 2. The minimum-overlap radius is zero and length of the network $X_{max} = 75$, width of the network $Y_{max} = 45$, the offset values X_{offset} and Y_{offset} are equal to '1'.

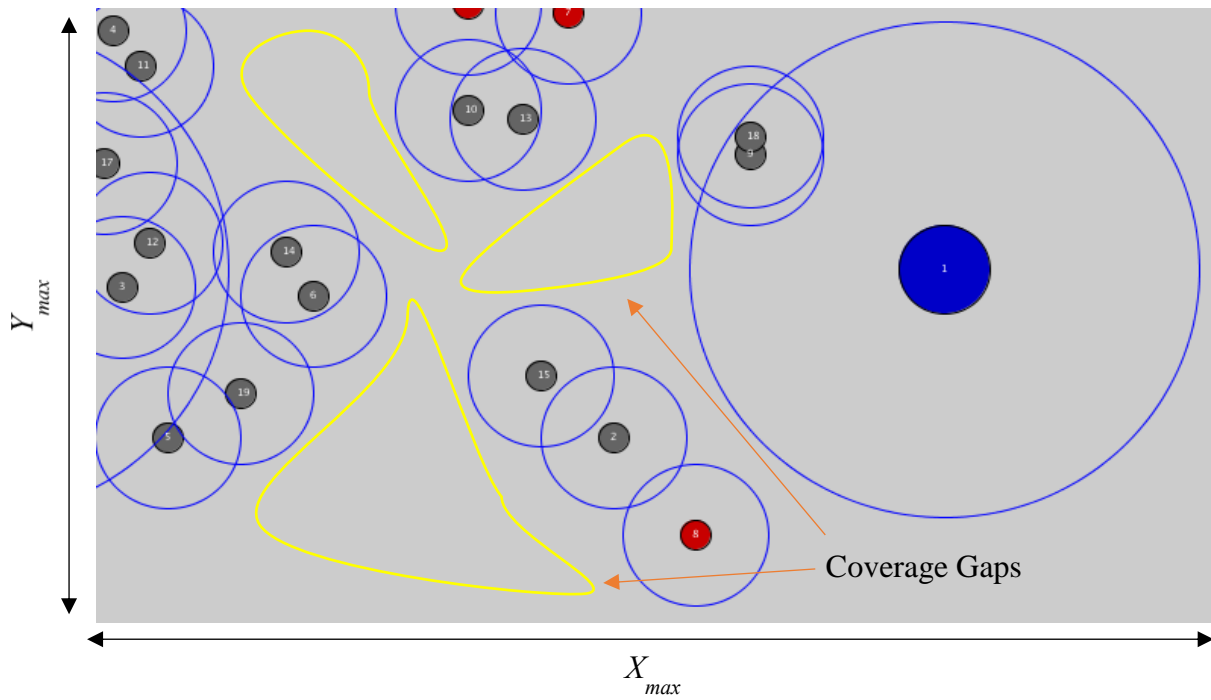


Figure 3.2. Network with $O_R = 0$, $V = 20$, $X_{max} = 75$, $Y_{max} = 45$, $X_{offset} = 1$, $Y_{offset} = 1$

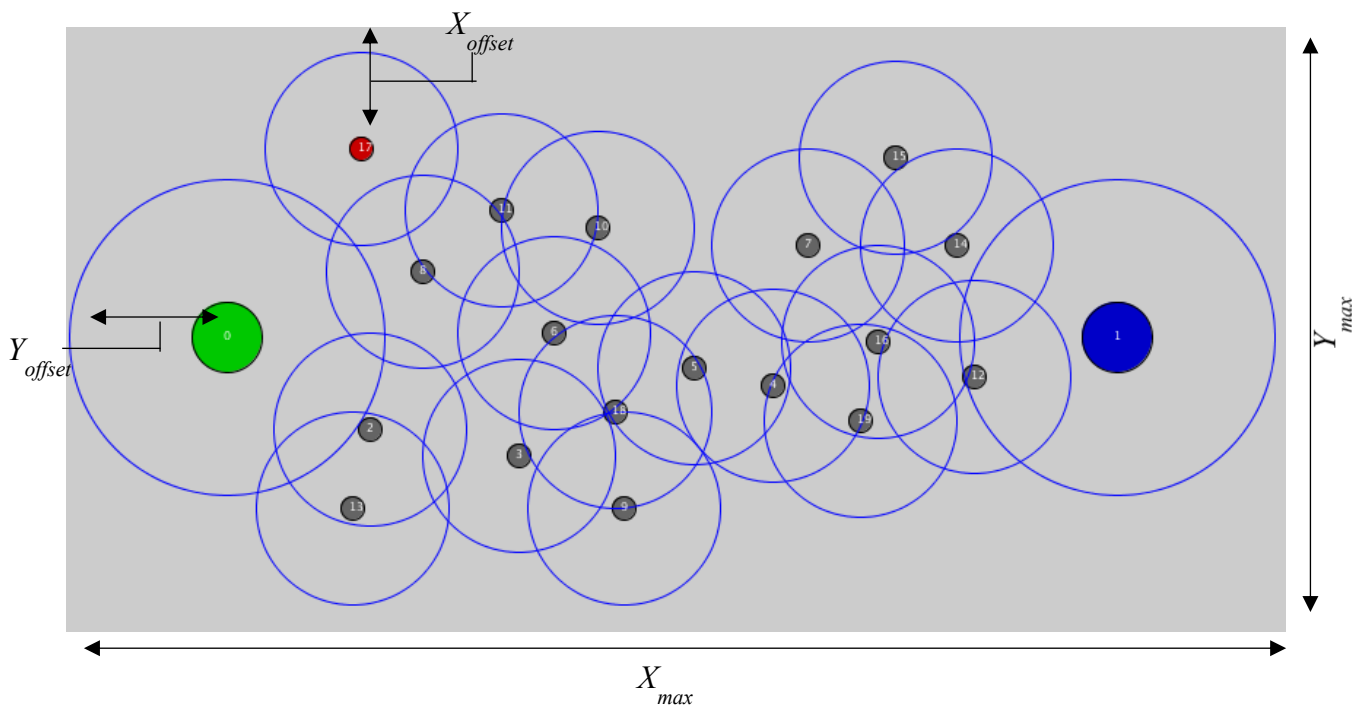


Figure 3.3 Network with $O_R = 5$, $X_{offset} = 25$, $Y_{offset} = 15$

From the figure 3.2, following observations were made

1. In the figure that only 16 nodes are clearly visible, as they deployed within the region.
2. Source node 0 has taken coordinates, which are outside the limits of the network.
3. More than one set of nodes are deployed very close to each other, for example consider node 9 and node 18, node 10 and node 13, node 4 and node 11, node 6 and node 14, node 3 and node 12.
4. Without the constraints there exists more than coverage gaps, also referred as coverage holes in the network.
5. There are three isolated nodes, highlighted in red. The process of identifying isolated nodes is explained later in this chapter.
6. Node 3, 12 and 17 are deployed very close the source node 0.

Note: Traditional methods follow a rule that if range circles of two nodes overlap then they are neighbors, which is not practically possible. Therefore, in this algorithm, an isolated node is one which is not in the range of antenna of other node.

Redrawing the figure 3.2 with constraints: By applying constraints with $O_R = 5$, $X_{offset} = 15$, $Y_{offset} = 25$, rest all parameters unchanged, following improvements are observed

1. All 20 nodes are visible in the region and no node is deployed beyond the limits of the region.
2. Source node and sink node both are deployed within the region of interest.
3. No two nodes are close to each other, all the nodes are evenly deployed throughout the network
4. Coverage gaps are removed and the deployed nodes provides complete coverage.
5. Isolated nodes are reduced from 3 to 1.
6. No node is deployed close to source node and sink node.

It is clear from figure 3.2 and 3.3 that applying the constraint gaurantees even random deployment of nodes so that no node is lost by getting depoyed beyond the region. The coverage is increased as there are no coverage gaps. The isolated nodes are also reduced which is again facilitating the enhanced coverage.

Note: Node position set is the novel technique which is implemented for the first time. No other work in the literature has ever reported such steps to be followed before deployment. Therefore

this is the proof for the claim made in section 2.9 that MNORDA is a pre-deployment technique.

3.5 Find Neighbor Algorithm

Once the node position set is executed, next step is to find neighbours in the network. For this the steps followed by each node are

1. **Step1 - Call inRange** : Every node tries to find a neighbour that lies within its communication range. To know which node is its neighbour, the current node calls the inRange function described in section 3.2.1. To simplify the representations, re-write the Eq 3.1 and 3.2 in terms of distance as follows:

$$d(S_m, S_n) = (\text{abs}(S_m.x - S_n.x) \leq S_n.\text{range} \ \&\& \ \text{abs}(S_m.x - S_n.x) \leq S_m.\text{range}) \quad \dots(3.6)$$

$$d(S_m, S_n) = (\text{abs}(S_m.y - S_n.y) \leq S_n.\text{range} \ \&\& \ \text{abs}(S_m.y - S_n.y) \leq S_m.\text{range}) \quad \dots(3.7)$$

Note: x and y coordinates of current node S_m and neighbor node S_n are compared with the range of both the nodes to prove that this algorithm is suitable for both homogenous nodes, where range of all nodes is same and for heterogenous networks where each node will have a different range.

2. **Step 2 - Send Hello Packet:** Once the inRange function is executed, next step to be followed is to find neighbors by calling the sendHello function. If the inRange function returns a true value for $d(S_m, S_n)$ given in Eq. 3.6 and 3.7, then a hello packet is sent from node S_m node S_n as given in Eq. 3.8

$$\text{hello} = \begin{cases} 1 & d(s_m, s_n) \leq R \\ \text{Null} & \text{Otherwise} \end{cases} \quad \dots (3.8)$$

Eq. 3.8 holds true for source node and sink nodes also. Therefore re-writing the Eq. 3.8 and given in Eq. 3.9 and 3.10.

$$\text{hello} = \begin{cases} 1 & d(s_l, \hat{s}) \leq R_s \\ \text{Null} & \text{Otherwise} \end{cases} \quad \dots(3.9)$$

$$\text{hello} = \begin{cases} 1 & d(t, s_r) \leq R_t \\ \text{Null} & \text{Otherwise} \end{cases}$$

...(3.10)

Where:

\hat{s} – Source Node with range $R_s \gg R$

t – Sink Node with range $R_t = R_s$

S_l – set sensors that exists on left boundary of the area

S_r – set of sensors that exists on right boundary of the area.

$d(\hat{s}, S_l)$ - Distance between node \hat{s} and set of nodes on left boundary

$d(s_m, s_n)$ - Distance between node S_m and node S_n

$d(t, S_r)$ - Distance between node t and nodes on right boundary.

- 3. Step 3 - Generate Own Row:** The next step of find neighbour algorithm is to generate own row. Every node in the network maintains a row, elements of which corresponds to neighbours that are in range of current node. The number of entries in the row should be equal to number of nodes in the network. To generate a row, every node has to first call inRange function and then send a hello packet to its neighbours. If a node responds to the hello packet then that node will become neighbour of current node and a '1' will be entered in the row of current node at a place which corresponds to node id of neighbour. Consider the Eq. 3.11 given below

$$R_\alpha = \sum_{a=0}^V \sum_{b=0}^V x_{ab} = \begin{cases} 1, & a \neq b \ \forall d(s_m, s_n) \leq R \\ 0, & a = b \end{cases} \quad \dots(3.11)$$

$\alpha = 0$ to V

Where:

R_α – Row of current node

x_{ab} - Elements of Row of current node

Elements of row will be '1' for the nodes which are in range of current node and '0' for all other nodes.

Note: For a element of row to be 1, two conditions are checked, one is the distance between two nodes should be less than their range, and the node should not be itself. The second avoids redundant packets in the network as a node will never send a packet to itself.

Example 2: To explain the practicality of find neighbour algorithm consider a network with 8 nodes as given in figure 3.4. After applying the find neighbour algorithm to the given figure, the row generated for a node as per Eq. 3.11, say for node 3 will be as follows:

$$R_3 = [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$$

As per figure 3.4, node 3 is in the range of node 2, node 4 and node 6. Therefore, in the row of node 3, a '1' is entered in place number 2, 4 and 6 and remaining elements are all 0's. In the same way, every other generates its own row, elements of which indicate the nodes that are in its direct range.

4. Step 4 - Share Row: After generating own row, every node shares its row with the immediate neighbours. Before sharing, to optimize the data packets and to avoid any extra overhead packets in the network, two checks are performed which are

Check 1: $S_m.hello[nodeid][i] == 1$, this check is to ensure that the node with which current node shares its information is its neighbour.

Check 2: $nodeid \neq nodearray[i].nodeid$, this check ensures that the node under consideration is not the node itself.

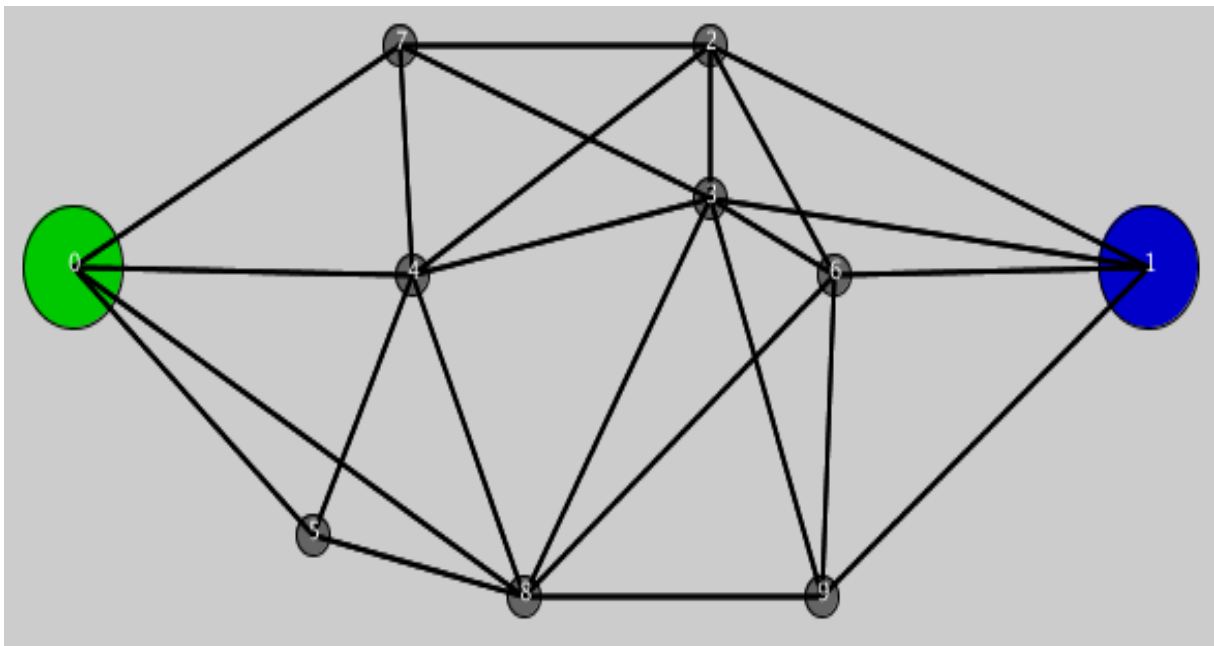


Figure 3.4 Example network with 10 nodes.

3.6 Tell-Neighbor Algorithm

In tell-neighbour algorithm, the information available with a node is shared with its neighbours. To implement this step spin function is called. The detailed explanation of execution of spin function is given below:

1. Spin function is called by every node after executing find neighbour algorithm.
2. Spin function shares the information of neighbors with neighbours, there by helping in spreading the information through out the network.
3. Spin function shares the rows among neighbouring nodes for generating adjacency matrix.
4. It is an recursive function, and is called again and again by every node until a node gets information of every other node in its adjacency matrix.
5. Order of adjacency matrix should be $V \times V$.
6. Spin function is called by a node and executed until the order of its adjacency matrix is $V \times V$.
7. To optimize the data packets in the network, spin function of this algorithm implements a one-way traffic where as the traditional method needs a two-way traffic.
8. For example, consider figure 3.4, in this node 2 and node 3 are direct neighbours and therefore will exchange the information of their rows with each other.
9. Node 6 in figure 3.4 is not in range of node 2, the row available with node 6 can be shared with node 2 through node 3, node 8 or node 9 as node 6 is in direct range of these three nodes.
10. The traditional method followed is that node 2 send a request packet to node 3 for sharing information of node 6. Then node 3 will send a reply to this request along with row information of node 6. This will create a two-way traffic in the network, one the request packet and second the reply packet.
11. As node 6 information can be shared with node 2 by node 3, node 8 or node 9, there is every chance that duplicate packets will be created in the network, which will any how be discarded by node 2.
12. To avoid the above mentioned problems, spin function in tell-neighbour algorithm implements one-way traffic by sharing the information of neighbours with neighbours, before they request and if and only if the neighbouring node doesn't have it.

13. To implement this one-way traffic technique the following check is performed
Check : if $(S_m.Row[i] == 1 \ \&\& \ S_n.Row[i] == 0)$: This check is to ensure that current node S_m has information of $Row[i]$ and its neighboring node S_n doesn't have the information of same $Row[i]$.
14. Then $S_n.Row[i] = S_m.Row[i]$, which means that $Row[i]$ from node S_m is shared with node S_n and the variable $S_n.rowHave++$ is updated accordingly. This variable represents the number of nodes information the current nodes has.
15. Spin function is called by current node S_n until the value of variable $S_n.rowHave++$ is equal to number of nodes in the network.

3.7 Generate Adjacency Matrix

Adjacency matrix of each node represents the information of total network. Some important properties of adjacency matrix are

1. The order of adjacency matrix will be always $V \times V$, where V is total number of nodes in the network.
2. Adjacency matrix will always be a square matrix.
3. Each row of adjacency matrix represents the information of each node, for example, row 0 represents information of node 0 and row 1 of node 1 and so on.
4. As a node never transmits any data packet to itself, the diagonal elements of adjacency matrix will have 0's
5. For Omni-Directional nodes, adjacency matrix will be symmetric.
6. Once every node has the same adjacency matrix, then it means that network is converged and there is no need to share any more information between the nodes.

Consider the example network given in figure 3.4, the adjacency matrix of each node, after the execution of tell-neighbor algorithm will be as given below.

$$Adjacency\ Matrix = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Note: At the end of tell-neighbor algorithm, every node should have same adjacency matrix. If any node has a different adjacency matrix then it means that there either is a packet loss, communication error or may even be the case where a node is dead. In such cases, symmetricity property of adjacency matrix can be used to implement interpolation.

3.8 Implementing Symmetricity in adjacency matrix

As the nodes considered here are Omni-Directional, consider the following points

1. If node m and node n are in range, then the row of node m will have 1 at node n 's place.
2. As per symmetricity row of node n should have 1 at node m 's place.
3. But in real time it may happen that packet from node m has reached node n but the reverse did not happen.
4. Reason may be packet loss, signal loss, error in communication.
5. In such cases row of node m will have 1 at node n 's place but row of node n will have 0 at node m 's place.
6. To avoid such ambiguity interpolation technique is implemented by taking the transpose of corresponding row and copying it to the corresponding column.

Consider the adjacency matrix given below, which is the error version of adjacency matrix given in section 3.7. The errors are represented in red entries.

$$Adjacency\ Matrix = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

As per the figure, row 2 has 0's at 2nd and 3rd place which means that node 1 is not in range of node 2 and node 3. But the same is not true in row 2 and row 3 as they have 1 at node 1's place. This may have happened due any one of the errors mentioned above. Same ambiguity can be seen in row 7, row 8 and row 9. To remove such error, interpolation is applied by taking

symmetry of the corresponding columns and pasting them in the corresponding rows. After interpolation, the same matrix is redrawn below.

$$Adjacency\ Matrix = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

3.9 Construct Barrier Paths

To construct k-barrier paths in the network, the communication range of successive nodes should overlap from one end of the network to the other end.

1. Requirement of k-barrier coverage is have at-least l -barrier coverage in the region of interest, i.e., minimum value of k should be 1.
2. Barrier paths are formed to facilitate multi-hop communication.
3. Sensors in barrier paths should be able to transfer the sensed data outside the network through multi-hop communication.
4. One of the goals of this research work is to maximize the value of k in k-barrier coverage, i.e., to form as many number of barrier paths as possible.
5. A network with V nodes can provide k-barrier coverage, if and only if, there exists disjoint-paths, in terms of node or edge, between source node \hat{s} and sink node t .
6. Consider $P = \{p_1, p_2, \dots, p_n\}$ as the set of paths between source node \hat{s} and sink node t , consisting of set of nodes from $S = \{\hat{s}, s_1, s_2, \dots, t\}$
7. If P_i is the i^{th} barrier path from the set P having N sensors from set S , then the definition of k-barrier coverage will be as given in Eq. 3.12

$$P_i = \bigcup_{i=1}^N S_i \quad \dots(3.12)$$

8. To forma a barrier path, the constraints to be followed are

Constraint 1: The Euclidean distance between two nodes S_m and S_n should be less than or equal to the range of nodes as given in Eq. 3.13

$$d(s_m, s_n) \leq R \quad \dots (3.13)$$

Constraint 2: Distance between source node \hat{s} and set of nodes S_l on the left boundary of the network should be less than or equal to the range of source node R_s as given in Eq. 3.14.

$$d(\hat{s}, s_l) \leq R_s \quad \dots (3.14)$$

Constraint 3: Distance between sink node t and set of nodes S_r on the right boundary of the network should be less than or equal to the range of sink node R_t as given in Eq. 3.15.

$$d(s_r, t) \leq R_t \quad \dots (3.15)$$

9. If the constraints defined from Eq. 3.13 to 3.15 is followed then a path P_i will exist from source to sink node as given in Eq. 3.16

$$P_i = \begin{cases} 1 & d(\hat{s}, s_l) \leq R_s \wedge d(s_m, s_n) \leq R \wedge d(t, s_r) \leq R_t \\ 0 & \text{Otherwise} \end{cases} \quad \dots (3.16)$$

10. Each barrier path P_i consists of set of sensor nodes denoted by v_s from S . These set of sensors participate in forming the k -barrier paths in the network.

Example 3: To form a barrier path from source to sink node, consider a network with 20 nodes. First, a demonstration of importance of constraints defined in Eq. 3.3 to 3.5 is provided by considering figures 3.5 and 3.6 given below:

1. In figure 3.5, length L of the network $X_{max} = 75$, $Y_{max} = 45$, Minimum-Overlap Radius $O_R = 0$, $X_{offset} = 10$ and $Y_{offset} = 10$. As $O_R = 0$, nodes are deployed very closed to each other as a result of which huge coverage gaps are created in the network at many places. The source and sink nodes are totally disconnected and therefore the number of barrier paths that can be formed is 0. It means value of $k=0$.

2. Figure 3.6 has the same value of X_{max} , Y_{max} and X_{offset} , Y_{offset} but the value of Minimum-Overlap Radius $O_R = 5$. Because of which there is a minimum separation distance of 5 units between the coordinated of two nodes.
3. Therefore, the nodes are evenly distributed throughout the network reducing the coverage gaps and thereby increasing the overall coverage of the network.
4. As the coverage is increased, one can find that source and sink nodes are now connected and barrier path can be formed between the two by connecting successive sensors from left boundary to right boundary. One such path formation is shown in the figure 3.6

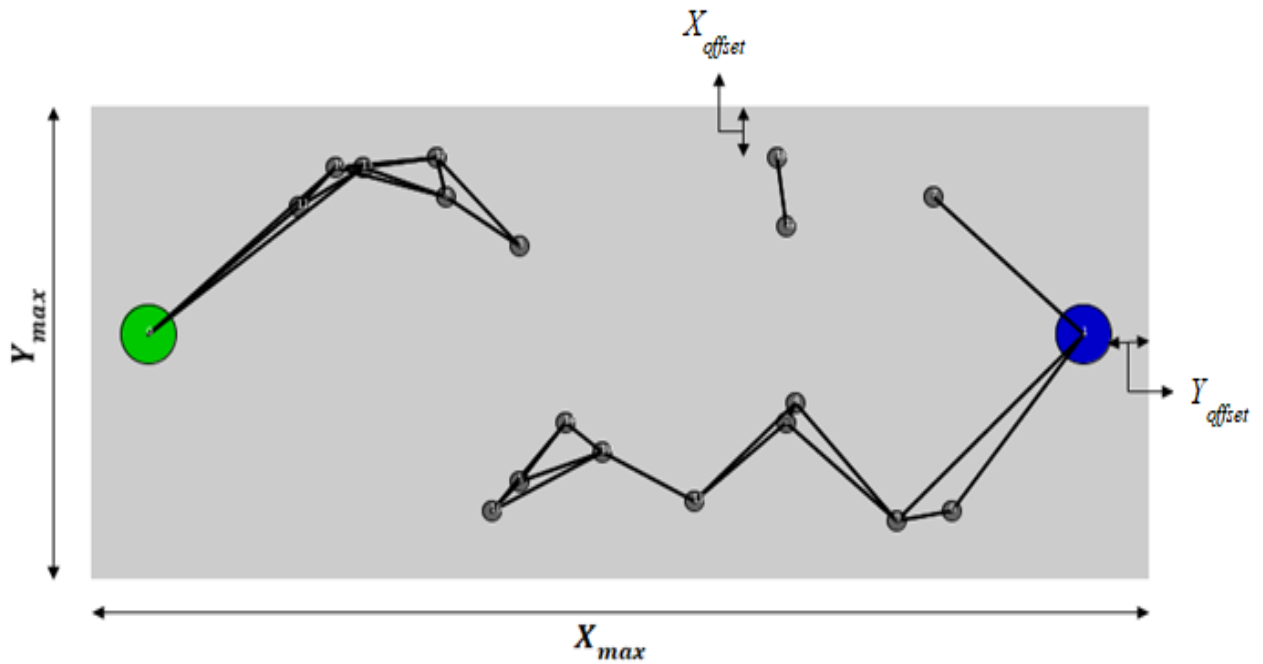


Figure 3.5 Network with 20 nodes and $O_R = 0$, $X_{offset} = Y_{offset} = 10$

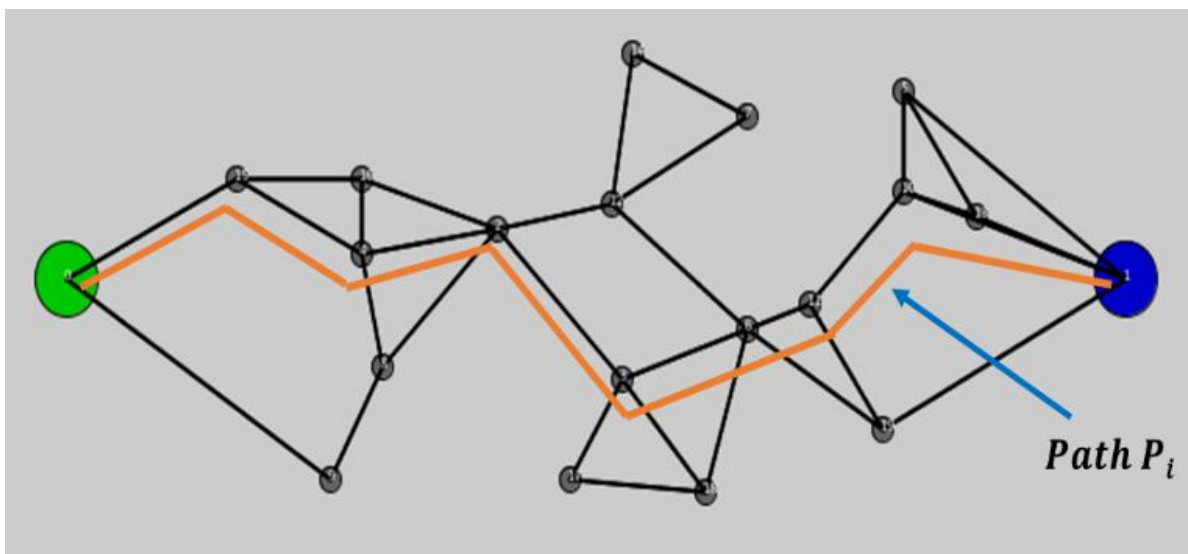


Figure 3.6 Network with 20 nodes and $O_R = 5$, $X_{offset} = Y_{offset} = 10$

3.9.1 Types of Barrier Paths: As stated earlier, types of barrier paths that can be formed are

1. **Edge-Disjoint Path:** An edge in the network is the straight line that exists between the two nodes that are in range. All the black line in the figure 3.6 are edges. An edge-disjoint path is one in which no edge can be shared by more than one path.

If v_s is the set of sensor nodes in the path P_i , then each node from set v_s can participate in more than one barrier path, but an edge can be shared by at most one path only, as given in Eq. 3.17

$$\sum_{\forall S} v_s \geq 1 \quad S \in V \quad \dots(3.17)$$

2. **Node-Disjoint Path:** A node-disjoint path is one in which no node can be shared by more than one path. Eq. 3.17 will be modified for node-disjoint as given in Eq. 3.18.

$$\sum_{\forall S} v_s \leq 1 \quad S \in V \quad \dots(3.18)$$

Consider the figure 3.6, the path P_i formed between source and sink node is node-disjoint path and there can be maximum one node-disjoint path in the network. Therefore the value of k for network of figure 3.6 is 1.

Now consider the figure 3.7 given below, which represents the same network as in figure 3.6, but in this case, edge-disjoint paths are drawn from source to sink nodes. One can observe that there are two edge-disjoint paths possible and therefore the value of k in this case is 2.

Therefore it is clear that edge-disjoint paths helps in maximizing the value of k in barrier covered Wireless Sensor Network and therefore in this thesis edge-disjoint paths are constructed. The proposed MNORDA algorithm can construct node-disjoint paths also.

3.9.2 Difference between traditional methods and MNORDA: While forming barrier paths in the network, traditional methods discussed in the literature given in chapter 2 follows a notion that if range of two nodes overlap then they are considered as neighbors.

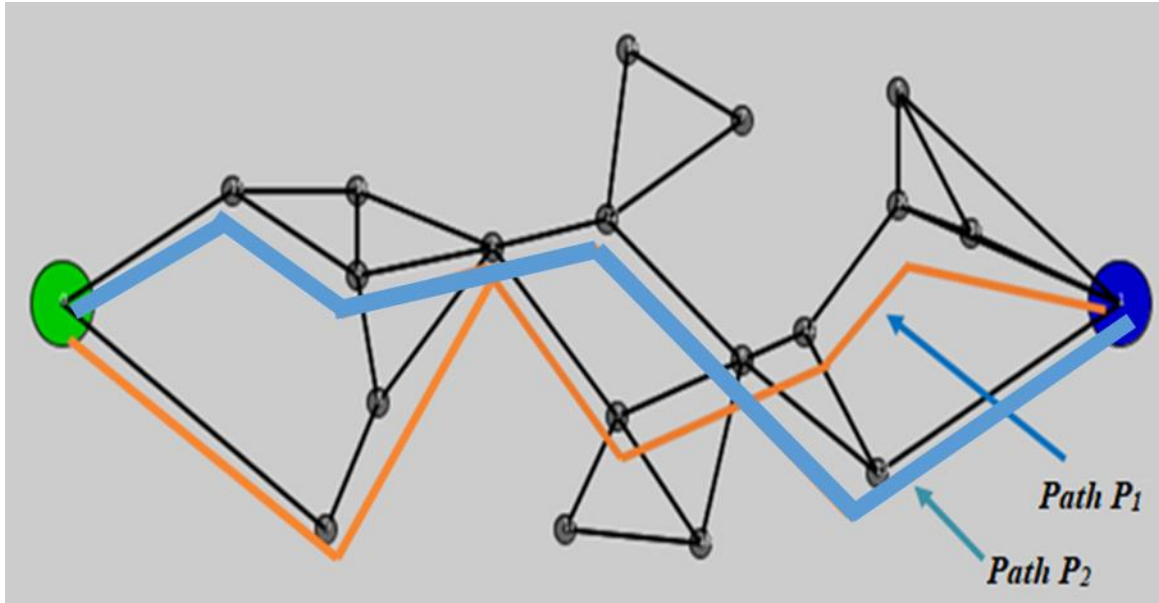


Figure 3.7 Network with two edge-disjoint paths.

In this case, if length of the region is L and communication range of each sensor node is R_s , then minimum number of nodes required to provide coverage is given by Eq. 3.19. The reference to Eq. 3.19 come from [54], where $\min(v_s)$ was mentioned as $\frac{L}{2R_s}$. The practicality of this assumption is difficult to prove since to provide barrier coverage, sensor range should overlap as shown in figure 3.8. By considering the overlap area as minimum value, +1 is added to Eq. 3.19. Consider the figure 3.8 given below where the length of the region is 20 units and communication range of each node is 2 units. Then as per the equation, minimum 6 nodes are required as given in figure 3.8

$$\min(v_s) = \left\lceil \frac{L}{2R_s} + 1 \right\rceil \quad \dots(3.19)$$

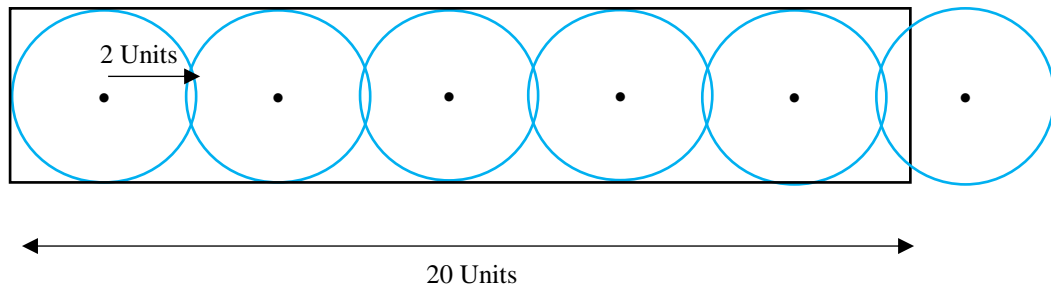


Figure 3.8 Number of sensors required in traditional methods

Note: In Eq. 3.19, as per traditional methods discussed in [54], number of sensors required are

exactly equal to $\frac{L}{2R_s}$, but in this thesis, it is assumed that one extra sensor is required to cover the gap created at the end of the network because of overlap of range circles of sensors.

In MNORDA algorithm, it is assumed that two nodes can communicate effectively if and only if range circles of antennas of two nodes overlap as shown in figure 3.9, which looks more practical than the method discussed in figure 3.8.

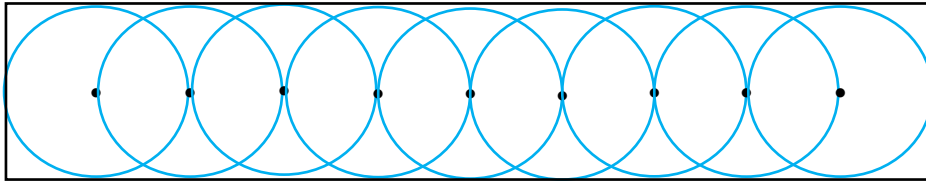


Figure 3.9 Number of sensors required in MNORDA algorithm

Therefore, in the proposed MNORDA algorithm to compute the exact number of sensors required, a variable \hat{L} is defined as the rearranged value of original length after adding the total overlap area of sensor nodes. Now to calculate number of sensors required Eq. 3.20 will be used, which is suitable for any size of overlap area.

$$\min(v_s) = \left\lceil \frac{\hat{L}}{2R_s} \right\rceil \quad \dots(3.20)$$

Where,

$$\hat{L} = L + \sum_{n=0}^{v_s-1} x_n$$

Where x is the size of overlap area of node n .

Note: Eq. 3.20 will work for random as well as sequential deployment and also for any value of x .

3.10 Identification of Isolated Nodes

To identify isolated nodes using the algorithm is a very easy task. Once the node-position set, find neighbor and tell-neighbor algorithms are executed then every node generates an adjacency matrix that represents the information of its immediate neighbors and also information of the complete network. If adjacency matrix of any node is having all 0 entries in rows and columns even after executing the three algorithms, then it can be treated as an isolated and disconnected node. In figure 3.2 and 3.3, isolated nodes are indicated in red color. Identification of these

isolated nodes reduces the power consumption of the network as these nodes can be permanently switched off and there by helps in increasing the overall lifetime of the network.

3.11 Identification of Shortest Path in terms of Distance and Number of Nodes

To identify the shortest path in the network, two parameters are considered, one is distance and the other is number of nodes. Shortest path in terms of distance is calculated in two steps

1. **Step 1:** By calculating the straight-line distance between the source node s and sink node t .
2. **Step 2:** By calculating the summation of distance between coordinates of successive nodes of all the paths.
3. **Step 3:** The path with distance closest to straight line distance between source and sink node is designated as the shortest path in the network.

3.11.1 Example Network: Consider figure 3.10 where three paths P_1, P_2, P_3 are shown. First the formulation of calculating shortest path between source node 0 and sink node 1 in terms of distance will be shown. It is obvious that, straight line distance is the shortest distance between any two points therefore to here first distance of each path formed will be compared with the straight line distance between node 0 and node 1. The path whose distance will be closest to the straight line distance will be considered as shortest path.

The stepwise process of finding the shortest path in terms of distance is shown below

1. To find shortest path, first calculate the distance of each path from node 0 to node 1
2. To find distance of each path, starting from node 0, find the number of nodes in each path.
3. Compute the distance between successive nodes of each path and add them.
4. The distance thus obtained will be the total distance of path, which can be compared with the straight-line distance between source and sink nodes and then path whose distance will be closest to straight-line distance, will be considered as shortest path.

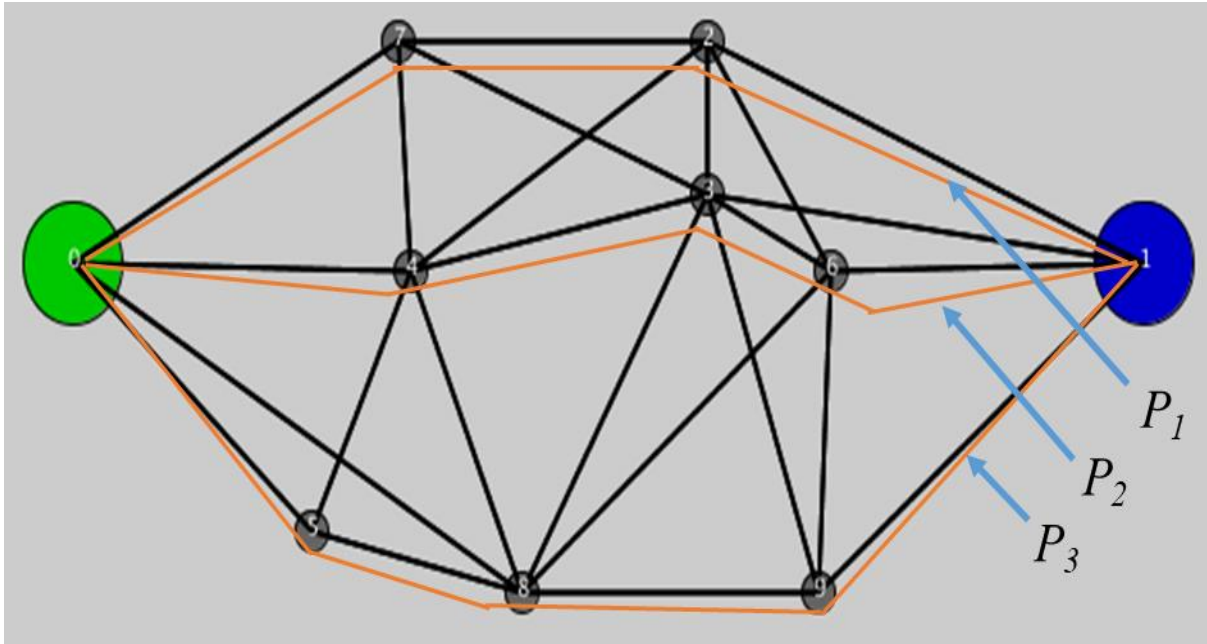


Figure 3.10 Network for calculating shortest path

- For example, in figure 3.10, consider path P_3 , to find the distance between coordinates of successive nodes use Eq. 3.21

$$P_3 = |d_{05}| + |d_{58}| + |d_{89}| + |d_{90}| \quad \dots (3.21)$$

Where d_{05} is the distance between node 0 and node 5 and so on.

5. Consider P_z as the z^{th} path to be formed and then Eq. 3.21 can be written in a generalized form as given in Eq. 3.22.

$$P_z = \left(\sum_{n=1}^{v_{sz}-1} |d_{qr}^n| \right) \quad \forall P_i \quad \dots(3.22)$$

Where $|d_{qr}|$ is the distance between node r and node q , given by the general straight-line equation as mentioned in Eq. 3.23

v_{sz} is set of nodes in path P_z^{th}

$$|d_{qr}| = \sqrt{(X_r - X_q)^2 + (Y_r - Y_q)^2} \quad \dots (3.23)$$

Where (X_r, Y_r) are coordinates of node r , (X_q, Y_q) are coordinates of node q

6. Applying the Eq. 3.22 to network given in figure 3.13 will give P_3 as the shortest path.

To identify shortest path in terms of number of nodes, the parameter considered is number of nodes traversed by the path including the source node but excluding the sink node. As defined earlier P_i is the set of paths having n nodes from v_s . Therefore, Eq. 3.22 will be used again, first to find out number of nodes in each path from the set P_i after which path with minimum number of nodes is computed which gives shortest path in terms of nodes as per Eq. 3.24

$$\min(v_{sz} - 1) \quad \forall P_z \quad \dots (3.24)$$

While computing number of nodes in a path sink node is not considered in the count of number of nodes as it is assumed that the all the paths ends at sink node, therefore in Eq. 3.24 while finding path with minimum nodes, 1 is subtracted from all the paths. In the path set P_z , with each path having v_{sz} nodes, Eq. 3.24 computes path with minimum nodes

Note: A practical network for showing the result of shortest path will be considered in the next chapter of simulation results. Shortest path in terms of nodes and in terms of distance will be shown.

CHAPTER 4 - SIMULATION RESULTS

Simulation experiments were performed under different conditions for verification of the endowed effectiveness of proposed Minimum Non-Overlap Radius Deployment Algorithm (MNORDA) algorithm. The empirical results thus obtained have been compared with the already existing state of art works to justify the superiority of the proposed algorithm.

The software used in this thesis for writing code of proposed algorithm MNORDA is termed as **Processing**, which is open source software shown in figure 4.1. It is graphical software best suitable for Wireless Sensor Networks. Version 3.3.7 is used to build the algorithm and perform the simulation.

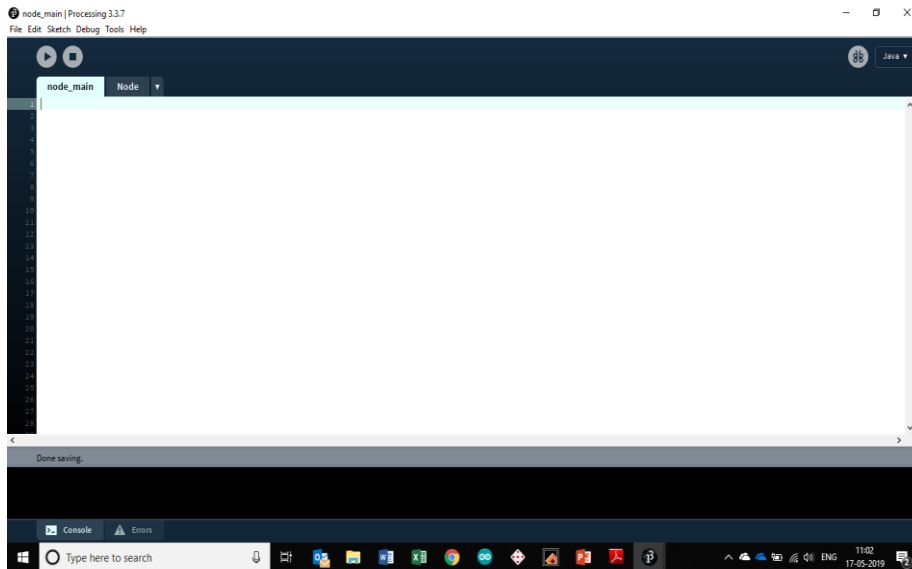


Figure 4.1 Processing software

Two main files have been used to write the code for complete algorithm

1. node_main: It is the area where the main simulator code for algorithm is written which renders and gives logic for the simulation environment.
2. node.pde: It is type of a library file which is used to define the blue print of node object and can be used for defining functions used in the main code.

4.1 Simulation result

To prove the effectiveness of Minimum Non-Overlap Radius Deployment Algorithm (MNORDA), simulation results are compared with the already existing state-of-art works for different parameters, viz.,

1. Number of barrier paths formed: To compare the number of barrier paths formed from one end of the network to the other end, the results of MNORDA is compared with

- Distributed Deployment Algorithm for Barrier Coverage (DDABC) proposed in [5]
- Centralized Barrier Coverage Algorithm based on Distributed Learning Automata (CBCDLA) given in [49]
- Autonomous deployment algorithm for barrier coverage with mobile sensors (MobiBar) proposed in [30]
- Approximate to Horizontal and Vertical Grid Barrier Algorithm (AHVGB) proposed in [28]

As per results given in figure 4.2, MNORDA outperforms the other algorithms in forming number of barriers for given number of nodes. The number of nodes considered for simulation are {100,200,300,400,500} for which number of barrier are obtained. The other parameters considered are, communication range of each node is 13, length of the network X_{max} is considered as 70, 100, 130, 150, 190 units, corresponding width of the network $Y_{max} = 50, 70, 90, 105, 135$ units, to incorporate 100, 200, 300, 400 and 500 nodes respectively. Minimum Non-Overlap radius for all cases is kept at 3 units.

The novelty of the proposed MNORDA algorithm is proved by the simulation results obtained where the number of barriers formed in this algorithm is more for different size of the network along with various number of nodes.

In table 4.1, number of barriers obtained for different node size is represented for all the algorithms. In comparison, the number of barriers obtained for MNORDA is more in number than other algorithms for all node size.

Table 4.1 Number of barriers for node size for all algorithms

Algorithms No. of Nodes	CBCDLA	AHVGB	MobiBar	DDABC	MNORDA
100	7	3	5	5	18
200	13	4	10	10	20
300	22	4	14	14	24
400	25	5	17	17	27
500	28	5	19	19	29

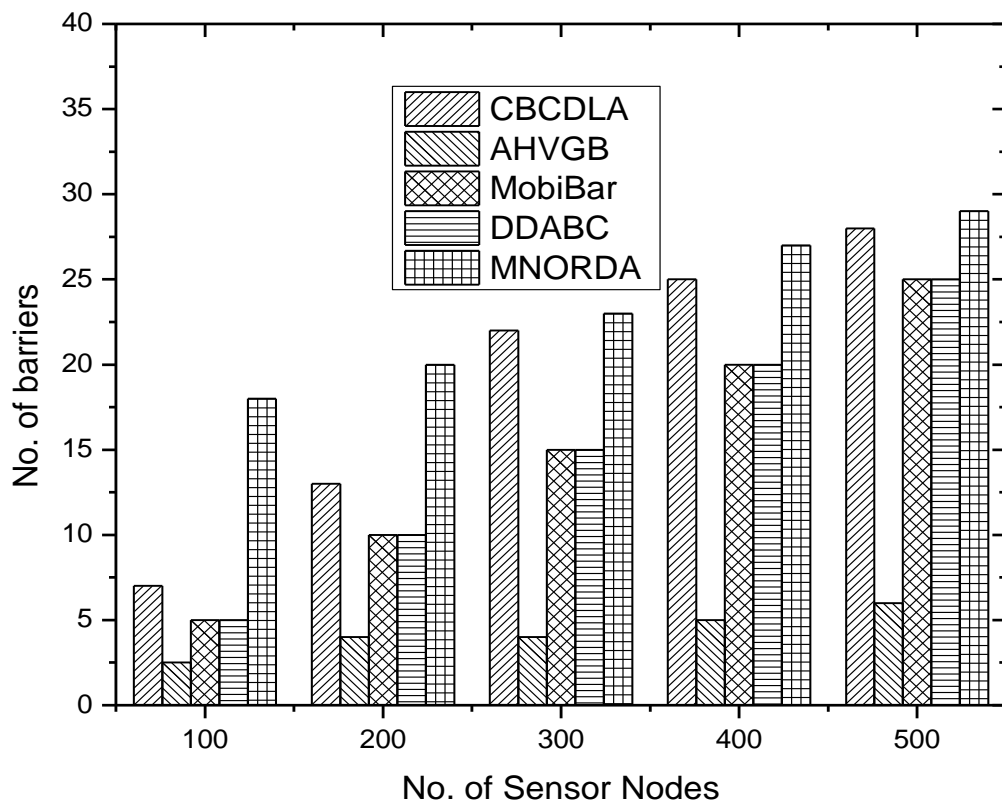


Figure 4.2 Number of Barriers vs Number of Sensor Nodes

As per the data given in table 4.1, MNORDA outperforms all the algorithms for all node size. The only algorithm which is close to the performance of MNORDA for node number 300 to 500 is CBCDLA.

Example 1: To prove that MNORDA is outperforming the other algorithms in terms of barrier formation below given are the results obtained by running the algorithm for 100 and 500 nodes.

Figure 4.3 shows 18 paths from source node to sink node, in red line and figure 4.4 show the corresponding output of the code indicating each of 18 paths along with length in terms of nodes and distance between source and sink node. While showing the length of each path in output, nodes closer to sink node will be displayed first and followed by nodes closer to source node. That is, the algorithm is showing nodes from left to right of the screen.

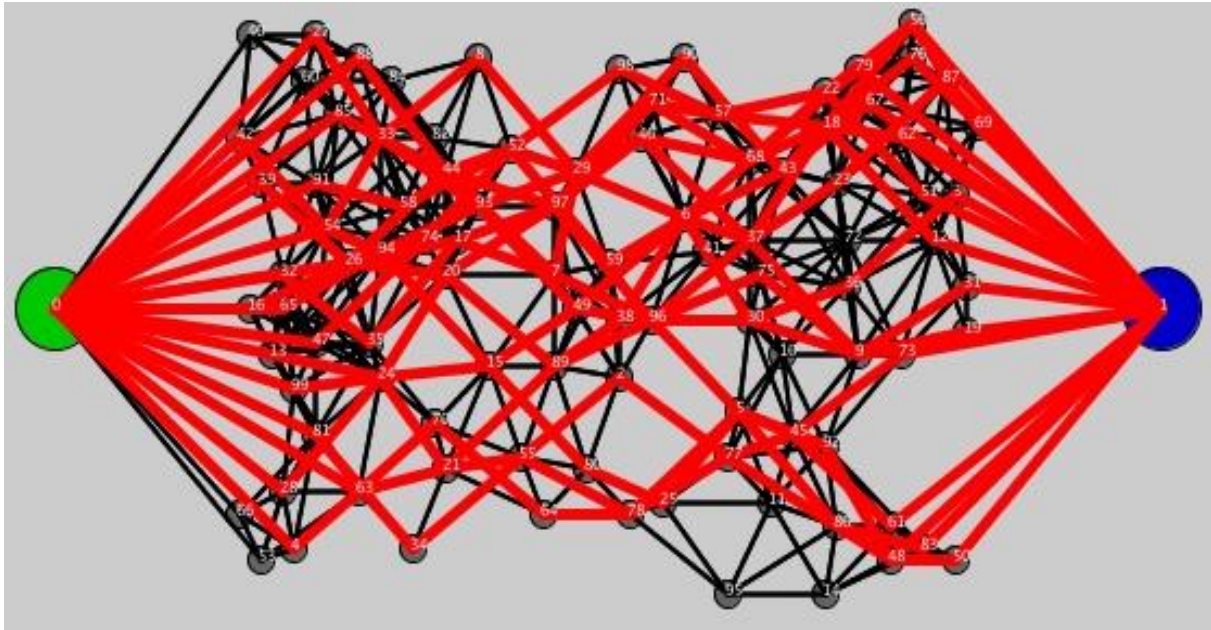


Figure 4.3 Network with 100 nodes and 18 barrier paths

```

node_main | Processing 3.3.7
File Edit Sketch Debug Tools Help

node_main Node
1 final int dia = 12; //diameter if the circles to eb drawn for each node
2 final int sdia = 3*dia; //scaled diameter for the source and sink nodes
3 final int xoffset = 150; //offset coordinates so that the nodes arent drawn too close to the edges of the screen
4 final int yoffset = 20; //offset coordinates so that the nodes arent drawn too close to the edges of the screen
5 final int scale = 5; //the scale factor for all the graphics on the screen, if less nodes keep higher value for more nodes keep lower, basically for zooming the graphics in or out

Starting iterations ran: 24
Algorithm 2 iteration number: 3
Sharing iterations ran: 2
Algorithm 2 iteration number: 4
Sharing iterations ran: 0
Algorithm 2 complete with overall 4 iterations
Algorithm 3 complete
Beginning node disjoint path discovery
There can be maximum 10 edge-disjoint paths from 0 to 1
Path 1 : 60 92 38 87 39 11 0 Len: 7 Distance: 497
Path 2 : 34 72 15 65 3 55 2 0 Len: 8 Distance: 458
Path 3 : 78 54 25 88 3 37 24 0 Len: 8 Distance: 443
Path 4 : 6 62 67 76 12 37 27 0 Len: 8 Distance: 455
Path 5 : 51 23 21 82 63 28 36 0 Len: 8 Distance: 422
Path 6 : 18 47 21 14 43 49 41 0 Len: 8 Distance: 436
Path 7 : 69 52 21 48 87 44 81 0 Len: 8 Distance: 468
Path 8 : 99 23 86 15 58 64 28 45 0 Len: 9 Distance: 439
Path 9 : 38 5 31 38 65 16 37 83 0 Len: 9 Distance: 432
Path 10 : 17 73 92 89 65 12 44 11 19 0 Len: 10 Distance: 489
Path 11 : 7 60 54 26 76 16 77 2 42 0 Len: 10 Distance: 508
Path 12 : 8 51 72 21 79 43 50 11 61 0 Len: 10 Distance: 516
Path 13 : 59 18 52 57 14 98 58 74 0 Len: 10 Distance: 462
Path 14 : 91 34 52 75 14 58 9 84 41 4 0 Len: 11 Distance: 532
Path 15 : 94 6 23 31 89 87 3 77 24 71 0 Len: 11 Distance: 515
Path 16 : 13 17 60 29 22 88 53 77 55 45 18 0 Len: 12 Distance: 595
Path 17 : 97 51 62 31 48 82 43 9 49 74 96 0 Len: 12 Distance: 549
Path 18 : 56 17 78 5 54 66 88 87 16 44 28 41 70 0 Len: 14 Distance: 661
0 Nodes are too close to each other

```

Figure 4.4 Result showing 18 disjoint paths along with length and distance

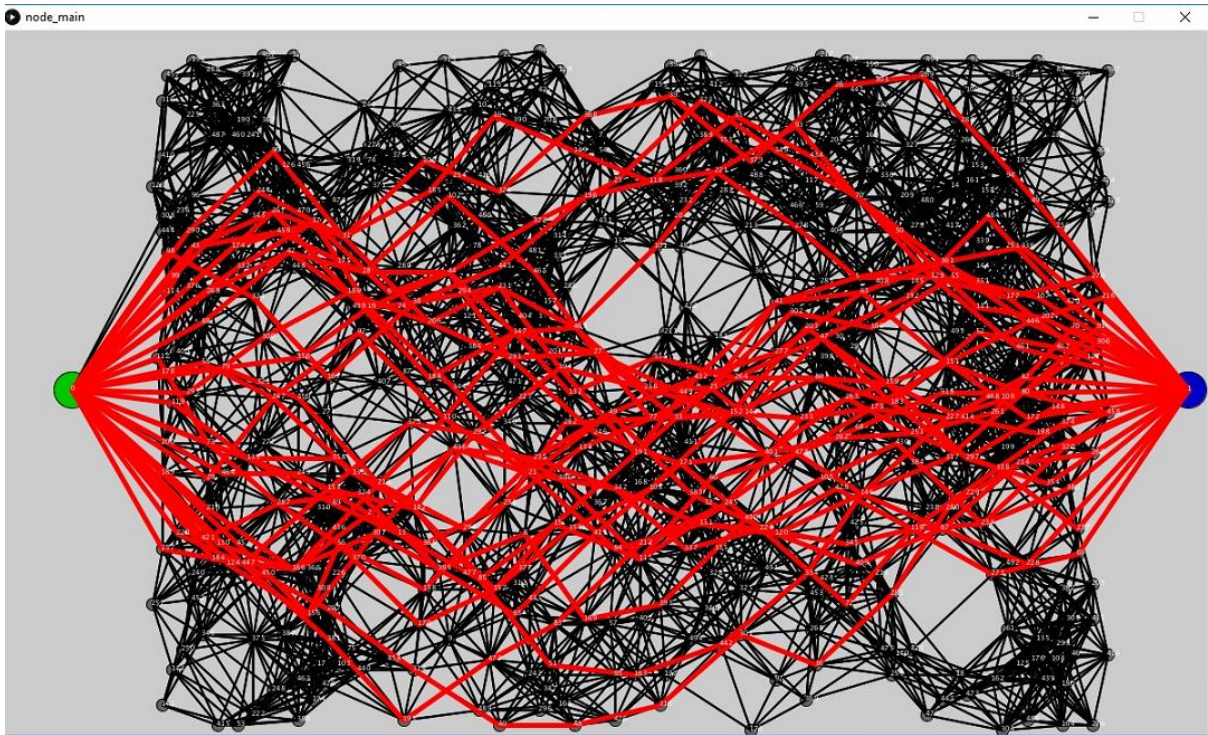


Figure 4.5 Network with 500 nodes and 29 disjoint paths

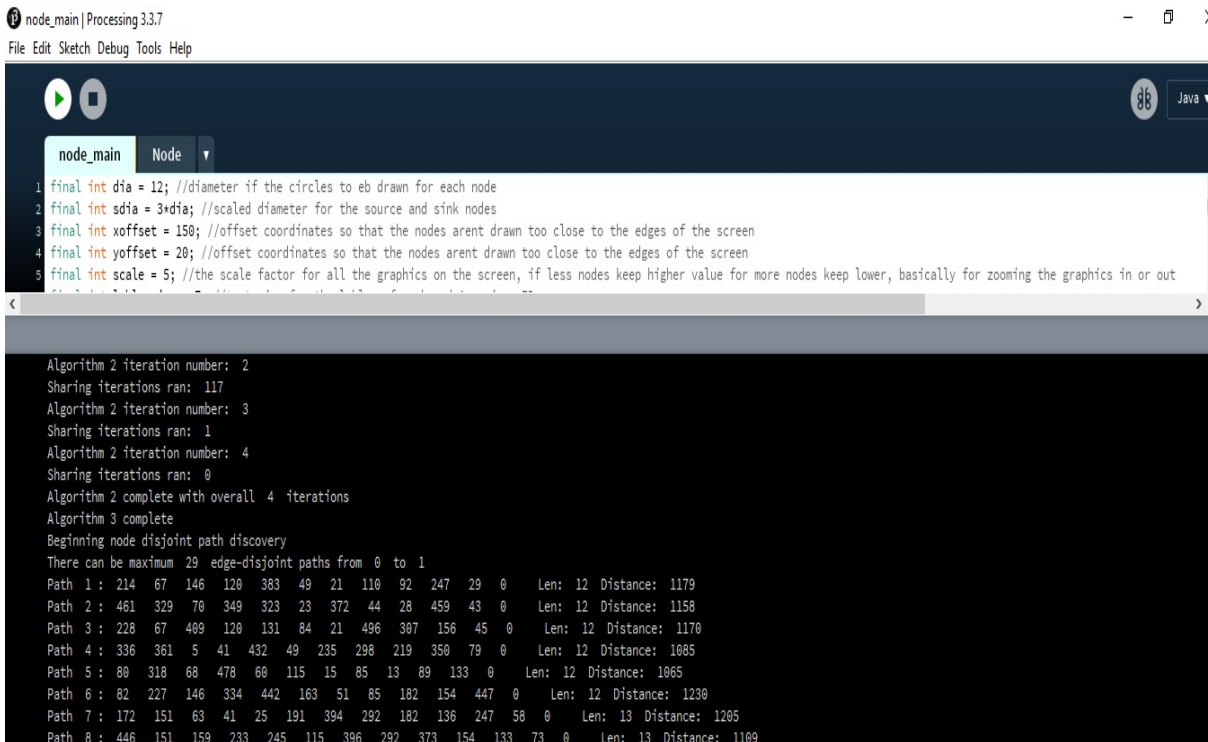


Figure 4.6 Result showing 29 disjoint paths along with length and distance

Figure 4.5 is for 500 nodes randomly deployed in the network and figure 4.6 is for showing that for 500 nodes 29 disjoint paths are formed in the network. MNORDA algorithm works

perfectly for 500 nodes also by deploying them uniformly throughout the network facilitating in maximizing the number of barrier paths between source and sink nodes. The constraints mentioned in equations 3.3 to 3.5 are the reasons for outperformance of MNORDA algorithm.

2. Energy Consumed: To compute the energy consumed while forming barrier paths in the network, number of packets exchanged between the sensor nodes is taken into consideration. To find neighbors, sensor nodes sends hello packet in the network and each packet exchanged needs certain amount power. Power consumed depends on the distance between two nodes and also on number of nodes in the network. As the number of nodes increases, number of barriers formed will also increase which will result in higher power consumption. To compare the performance of energy consumed of MNORDA, the algorithms picked from the literature are:

- Autonomous deployment algorithm for barrier coverage with mobile sensors (MobiBar) [30]
- Approximate to Horizontal and Vertical Grid Barrier Algorithm (AHVGB) [28]
- Minimax Algorithm [6]
- Virtual Force Based Algorithm [26]

As per figure 4.7, number of packets consumed by MNORDA is more than AHVGB, Minimax, and Mobibar but less than VF algorithm for node numbers 200, 300 and 400. When number of nodes are 100, MNORDA gives better performance than all algorithms except AHVGB. One of the obvious reason for more energy consumption is that the number of barriers formed is far more in MNORDA than the other algorithms and as stated more the number of barriers formed more is the data exchanged between the nodes. However, if there exist a threshold between number of barriers and energy consumed then MNORDA will give better performance than the other state of art works.

Note that, the results are plotted on logarithmic scale and therefore the energy consumed represented as number of packets is multiplied by 10000.

The network parameters i.e., communication range of node, length and width of the network are unchanged and are same as given for results obtained in figure 4.2

The total number of packets consumed by MobiBar, VF, MiniMax, AHVGB and MNORDA for different node size is given in table 4.2.

Table 4.2 Total number of Packets ($\times 10000$) Exchanged

Algorithms No. of Nodes	MobiBar	VF	MiniMax	AHVGB	MNORDA
100	5	5	1	0.5	0.97
200	5	9	1	0.5	4
300	5	11	1	0.3	9
400	5	50	1	0.2	16
500	28	5	19	19	29

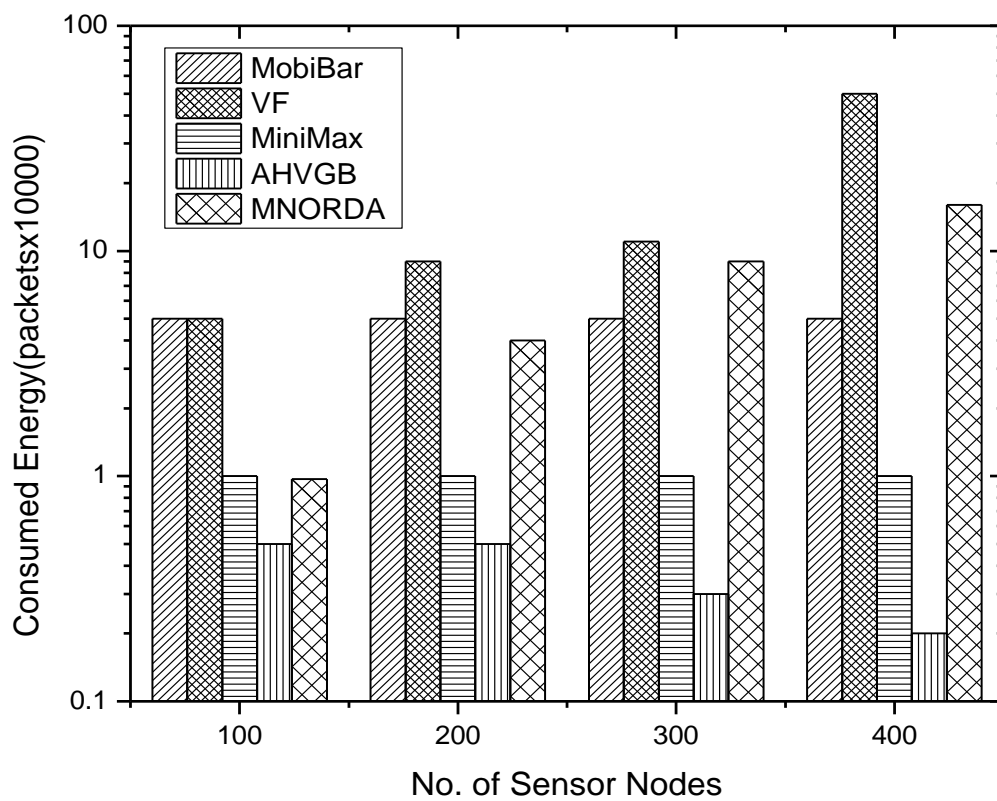


Figure 4.7 Consumed Energy vs Number of Sensor Nodes

Out of all the algorithms, Virtual-Force (VF) consumes more power than any other algorithm and AHVGB gives the best performance as it consumes very less power.

Example 2: To analyse the performance of MNORDA algorithm for number of packets consumed 100 nodes are considered in Figure 4.8 which gives the total number of packets exchanged between 100 nodes in the network for forming 18 barrier paths. As per the output in the figure a total of 9728 packets were exchanged.

```

node_main Node
1 final int dia = 14; //diameter if the circles to eb drawn for each node
2 final int sdia = 3*dia; //scaled diameter for the source and sink nodes
3 final int xoffset = 150; //offset coordinates so that the nodes arent drawn too close to the edges of the screen
4 final int yoffset = 20; //offset coordinates so that the nodes arent drawn too close to the edges of the screen
5 final int scale = 6; //the scale factor for all the graphics on the screen, if less nodes keep higher value for more nodes keep lower, basically for zooming the graphics in or out

Algorithm 2 complete with overall 3 iterations
Algorithm 3 complete
Beginning node disjoint path discovery
There can be maximum 18 edge-disjoint paths from 0 to 1
Path 1 : 15 2 9 31 10 0 Len: 6 Distance: 633
Path 2 : 23 7 9 82 16 0 Len: 6 Distance: 759
Path 3 : 35 36 9 72 22 0 Len: 6 Distance: 638
Path 4 : 96 36 95 31 33 0 Len: 6 Distance: 664
Path 5 : 58 2 26 45 77 0 Len: 6 Distance: 661
Path 6 : 6 30 43 45 87 0 Len: 6 Distance: 624
Path 7 : 8 30 26 28 14 21 0 Len: 7 Distance: 740
Path 8 : 5 40 65 82 22 32 0 Len: 7 Distance: 740
Path 9 : 11 20 13 37 56 47 0 Len: 7 Distance: 779
Path 10 : 24 30 67 45 10 50 0 Len: 7 Distance: 644
Path 11 : 48 49 26 4 19 51 0 Len: 7 Distance: 650
Path 12 : 68 44 13 93 34 73 0 Len: 7 Distance: 684
Path 13 : 12 20 91 37 19 81 0 Len: 7 Distance: 703
Path 14 : 42 39 97 55 19 83 0 Len: 7 Distance: 776
Path 15 : 69 39 91 93 29 89 0 Len: 7 Distance: 765
Path 16 : 27 41 91 98 29 22 71 0 Len: 8 Distance: 672
Path 17 : 59 35 54 9 64 77 85 0 Len: 8 Distance: 771
Path 18 : 66 96 54 95 94 22 86 0 Len: 8 Distance: 720
0 Nodes are too close to each other
Distance between source and sink nodes = 580
Execution time: 500 which is 0 mins 0 sec 500 ms
Source node: ( 170 , 70 )
Total Packets: 9728

```

Figure 4.8 Result showing packets exchanged for network with 100 nodes

3. Termination Time: Termination time can be defined as the time taken by the algorithm to terminate or time taken to get executed completely. It is also defined as convergence time i.e., the time taken by the network to get converged. Here in MNORDA converging of network involves six steps,

- To deploy sensor nodes randomly throughout the network to have a uniform coverage,
- To search for neighbors and update the list of neighbors with each node,
- To share information of neighbors with neighbors,
- To form k -barriers from source to sink nodes ($k \geq 1$),
- To identify isolated nodes,
- To find the shortest path in terms of number of nodes in that path and distance of path from source to sink nodes

To compare the performance in terms of termination time of MNORDA, the algorithms considered from literature are:

- Autonomous deployment algorithm for barrier coverage with mobile sensors (MobiBar) [30]
- Approximate to Horizontal and Vertical Grid Barrier Algorithm (AHVGB) [28]
- Minimax Algorithm [6]
- Virtual Force Based Algorithm [26]

Table 4.3 Termination time in seconds for all algorithms

Algorithms No. of Nodes	MobiBar	VF	MiniMax	AHVGB	MNORDA
100	1000	10000	100	85	0.5
200	1000	50000	100	85	3
300	1000	90000	100	85	11
400	1000	10000	100	85	37

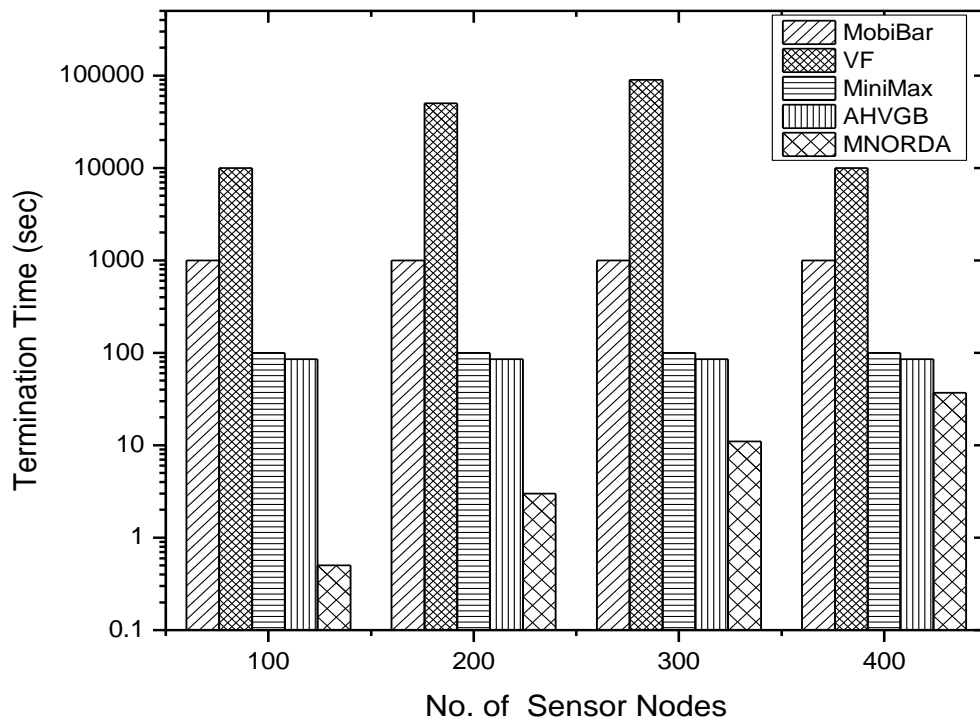


Figure 4.9 Termination time vs number of sensor nodes

Table 4.3 and figure 4.9 given above is to represent the time taken in seconds by MobiBar, VF, MiniMax, AHVGB and MNORDA algorithms to terminate in finite time and converge the network. As per the table MNORDA is the fastest of all and therefore will terminate and converge the network faster than the other algorithms for node size of 100, 200, 300 and 400. For node size greater than 400, MNORDA has proved to terminate in finite time. However, MNORDA will be faster for larger node size also.

4. Lifetime of Barriers: Lifetime of barriers refers to the time the network can provide k -barrier coverage continuously. Each barrier consists of sensor nodes which has limited battery and therefore can contribute to barrier formation during its entire lifetime. To compute the time for which the network can continuously provide barrier coverage, operating lifetime of each barrier is computed. As given in [54] the lifetime of barriers is computed in number of weeks. The algorithms considered for lifetime comparison of barriers are

- Imperialist Competitive Algorithm for Barrier Coverage (ICABC) [54].
- Two Round Maximum Flow Algorithm (TMFA) [48].
- Disjoint Path Algorithm (DPath) [47].

Table 4.4 Lifetime versus number of nodes for all algorithms

Algorithms No. of Nodes	Dpath	ICABC	TMFA	MNORDA
100	8	16	16	20
200	15	22	38	60
300	15	27	70	80
400	22	30	78	100
500	24	40	98	120

The results of MNORDA for barrier lifetime in weeks for different number of sensors nodes is given in table 4.4 and plotted in figure 4.10. As per the figure, for 100 nodes, lifetime provided by MNORDA is a little more than ICABC and TMFA algorithms. But as the node number increases to 200, 300, 400 and 500, MNORDA outperforms its counterparts ICABC, TMFA and Dpath algorithms by a big margin. Therefore, it is clear from these results that MNORDA algorithm has the best operating lifetime when compared to other algorithms.

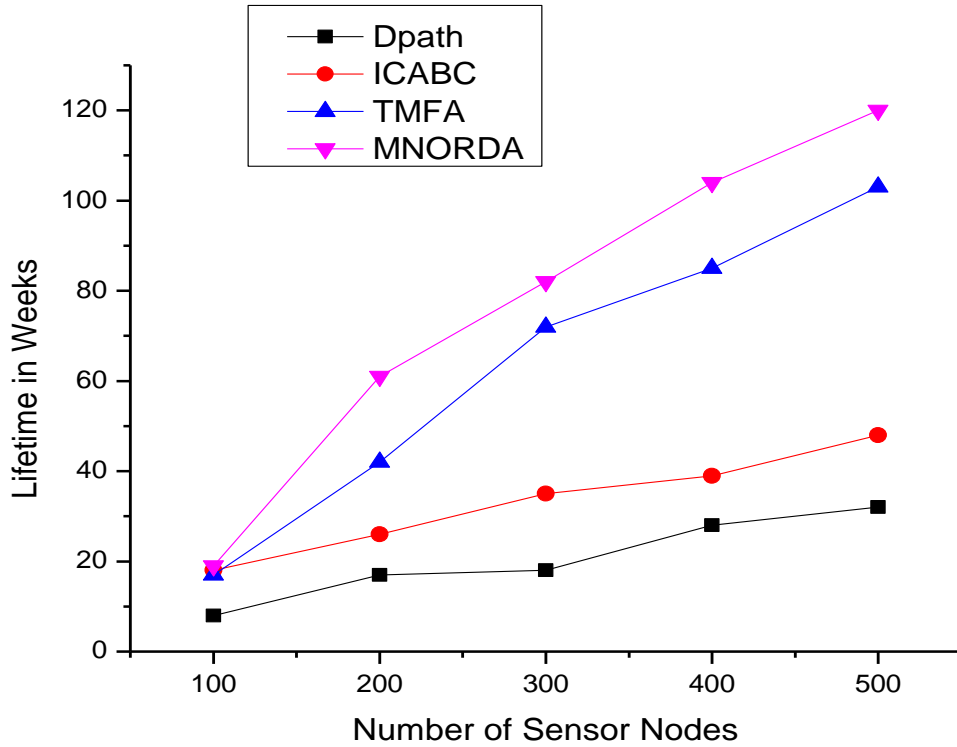


Figure 4.10 Lifetime of barrier in weeks

5. Sensor Utilization: Sensor utilization is defined as the ration of sensors participating in barrier formation to total number of sensors deployed. To compute and compare the sensor utilization factor for MNORDA algorithm, the research works from literature that are considered are:

1. Efficient k-barrier construction mechanism (EBCM) proposed in [62] includes best-fit coverage approach (BCA) and top-down one-coverage barrier approach (TOBA). The BCA approach includes three phases, in first phase the complexity of forming k -barrier coverage is reduced after which in second phase an attempt is made to maximize number of barriers formed and in third phase sleep wake-up scheduling technique is proposed to balance the overall energy consumption of the network. The BCA approach finds a node with a predefined weight value k . Depending on the value of weight k of each node, it is decided whether that node can participate in forming the k^{th} -path. If the weight value is less than k then that sensor cannot participate in k^{th} -path as a result of which the sensor utilization is reduced. To overcome this limitation, TOBA approach is proposed which does not depend on

- weight value k . But the problem associated with TOBA is that it can construct only 1-barrier at one time which means the convergence time of TOBA will be very high.
- Maximum Disjoint Path (MDP) algorithm proposed in [45] is a centralized algorithm which finds k disjoint paths between source and sink nodes. From the constructed k paths MDP finds the shortest path between two ends of the network. As MDP is a centralized algorithm it consumes high amount of power and also has downside in terms of scalability and flexibility

Table 4.4 Percentage of Sensor Utilization

Algorithms No. of Nodes	MDP	BCA	TOBA	MNORDA
300	40%	10%	20%	60%
350	45%	15%	29%	60%
400	50%	20%	35%	60%
450	52%	20%	40%	65%
500	59%	22%	50%	68%
550	69%	35%	61%	75%
600	81%	51%	80%	85%

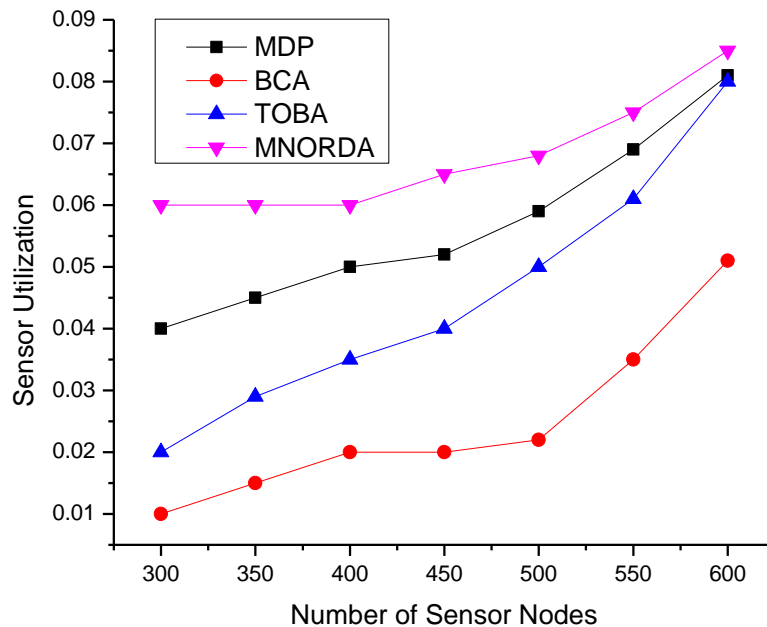


Figure 4.11 Sensor Utilization vs Number of Sensor Nodes

The simulation result for sensor utilization is given in figure 4.11 and data in table 4.4. As given in the figure the utilization of sensors in MNORDA is more than MDP, BCA and TOBA

algorithms. The sensor utilization in MNORDA is 60 % for 300 nodes and goes up to 85% when nodes are 600.

4.2 Performance of MNORDA for different values of O_R

Minimum Non-overlap radius O_R as described in the previous sections is the novel feature of proposed MNORDA algorithm. To prove the effectiveness of O_R , in this section few simulation results are shown for forming number of barriers for different node size. In each simulation, the value of O_R is varied and it will be proved that number of barriers formed increases while keeping all other parameters unchanged.

In Figure 4.12 O_R value considered is 3, 6 and 9 for node size of 100, 200, 300 and 400. As can be seen from the figure that as the value of O_R is increased the number of barriers increases for node numbers 100 to 300 but for node number 300 to 400 the number of barriers are same when value of O_R is 6 and 9. The reason is that the size of the network and the communication range of sensor nodes are unchanged and therefore the number of barriers is also the same. Therefore the number of barriers increases for the node size 100 to 400 when value of O_R is 3.

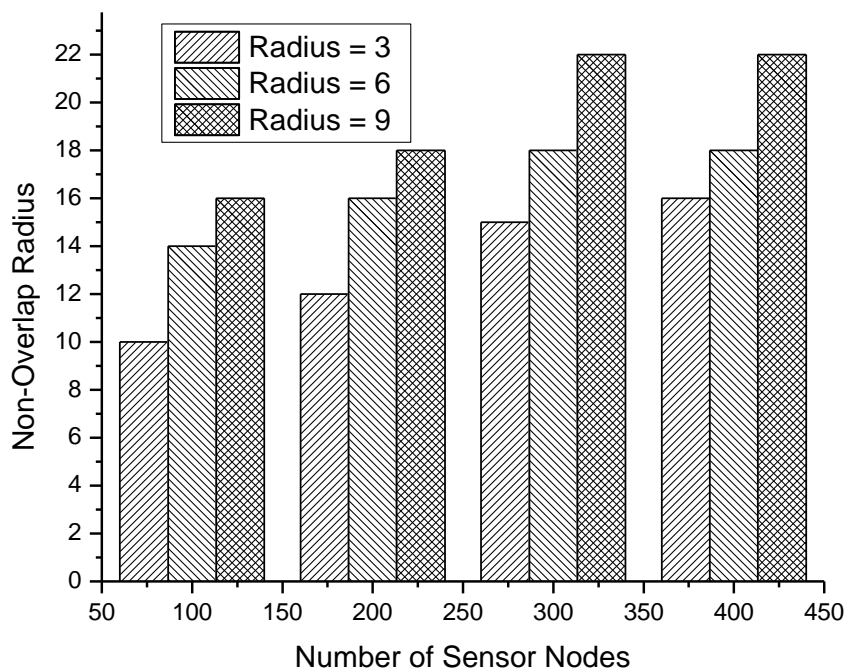


Figure 4.12 Effect of O_R on number of barriers

4.3 MNORDA Sample Result

As mentioned in section 3.3, MNORDA algorithm consists of 8 phases. In this section, a network with 10 nodes is considered to show the output after each phase of the algorithm. The output will have adjacency matrix after phase 1, i.e., node-position set, adjacency matrix after algorithm 1 and algorithm 2 i.e., find neighbour and tell-neighbour algorithms respectively, adjacency matrix before and after applying symmetry. The output will also have barrier paths, shortest paths in terms of nodes and distance and isolated nodes.

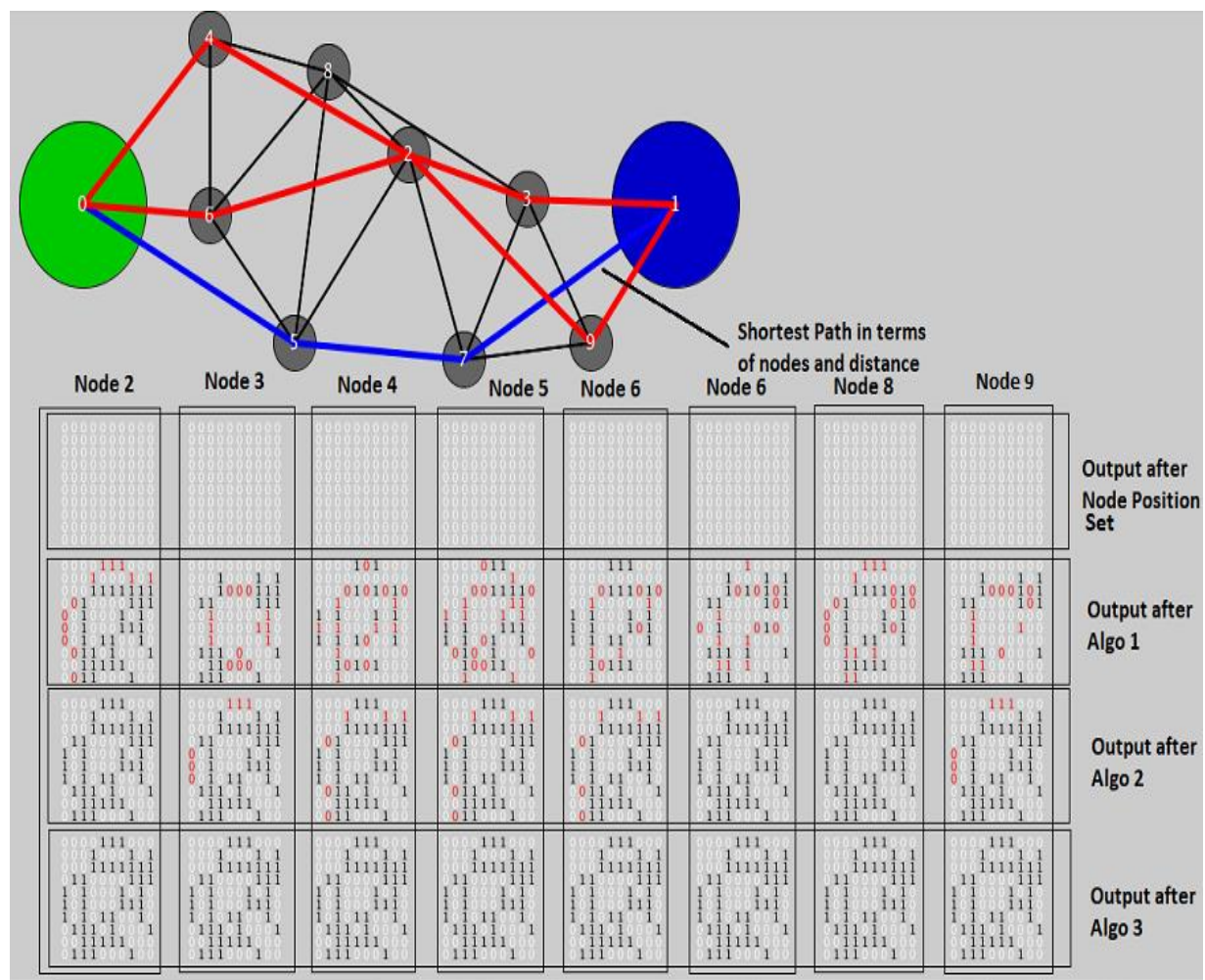


Figure 4.13 Sample result for MNORDA

Considering the result given in figure 4.13, the following points explain the result step-wise

1. Figure 4.13 consists of a network with 10 nodes and output in terms of adjacency matrices after node-position, algorithm 1 and algorithm 2 and also after implementing symmetricity.

2. Adjacency matrices of all nodes except source and sink nodes are shown
3. Adjacency matrix after node-position set of all nodes are zero because in this step the random deployment phase of MNORDA algorithm is carried where the nodes are uniformly deployed applying all the constraints mentioned in chapter 3. Therefore entries of adjacency matrix will all be 0's as the nodes have not yet started the communication.
4. Next shown in the figure is output after algorithm 1 which is find neighbor algorithm. The adjacency matrix here will have information of neighbors of current nodes plus the row information shared by neighbors of current nodes. As the network is small, every node is at a distance of 1-2 hops from the current node, therefore maximum information will be shared in this step only. As a result of which adjacency matrices of all nodes will have entries in all rows and columns.
5. Few entries in the matrices are in red in the outputs of algorithm 1 and algorithm 2. Each red entry represents the ambiguity mentioned in section 3.8. The number of red entries reduces from algorithm 1 to algorithm 2 for the obvious reasons as mentioned in the explanation of these algorithms.
6. After the output of algorithm 3 i.e., after implementing symmetry there are no red entries left and information of every node will be shared with every other node.
7. The output in the figure also has paths in different colors. Each color shown has a meaning with respect to shortest path. Blue color path represents the shortest path in terms of both distance and number of nodes.

Consider figures 4.14 and 4.15 given below. These two figures represent the computation of shortest paths in terms of distance and number of nodes for node size of 900 and 700 respectively. The adjacency matrix for each node will be of order 900x900 and 700x700 which is out of the scope of simulator screen to display the matrix of this size. Therefore in the output only nodes and paths are displayed.

In figure 4.14, paths with three different colours are visible. Yellow paths indicate shortest paths in terms of number of nodes. All the yellow paths have the same number of nodes which is equal to the minimum number of nodes in all the paths. Blue coloured path is a special case where a path consists of least number of nodes and also smallest distance between source and sink nodes. Red coloured paths are the other paths formed which do not satisfy least number of nodes and smallest distance.

In figure 4.15 a green path is visible which satisfy the minimum distance criteria but not minimum nodes criteria.

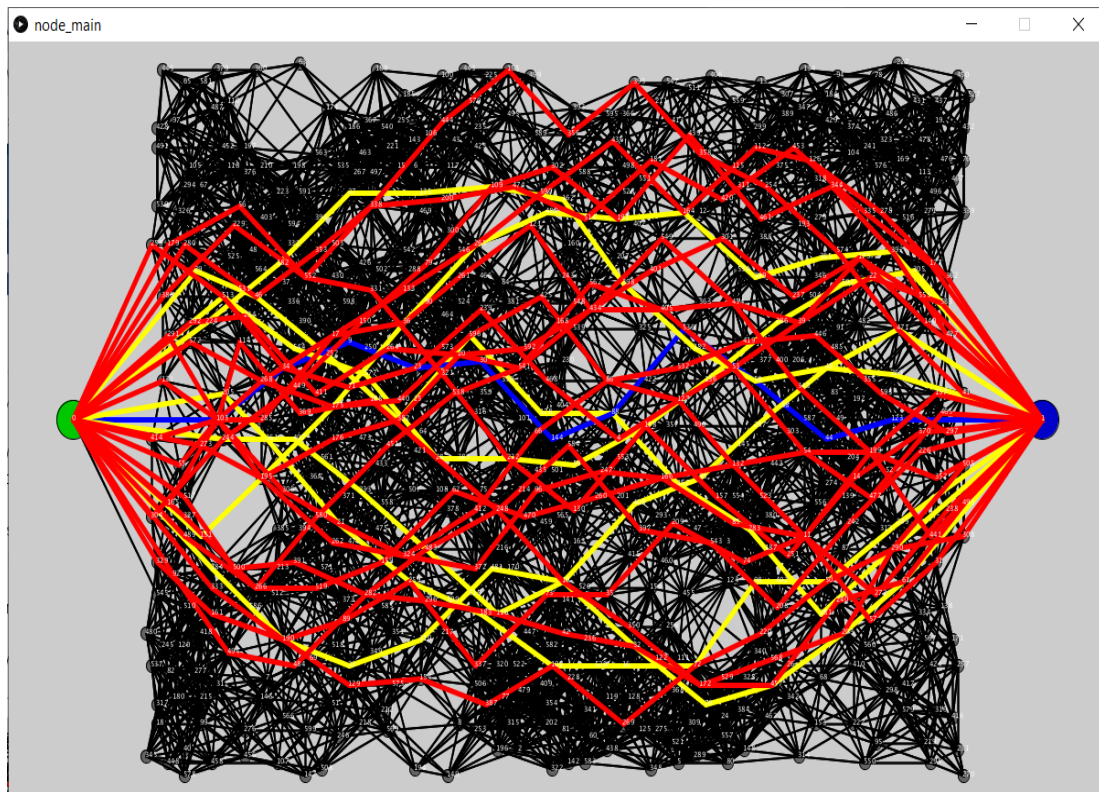


Figure 4.14 Paths for node size of 900

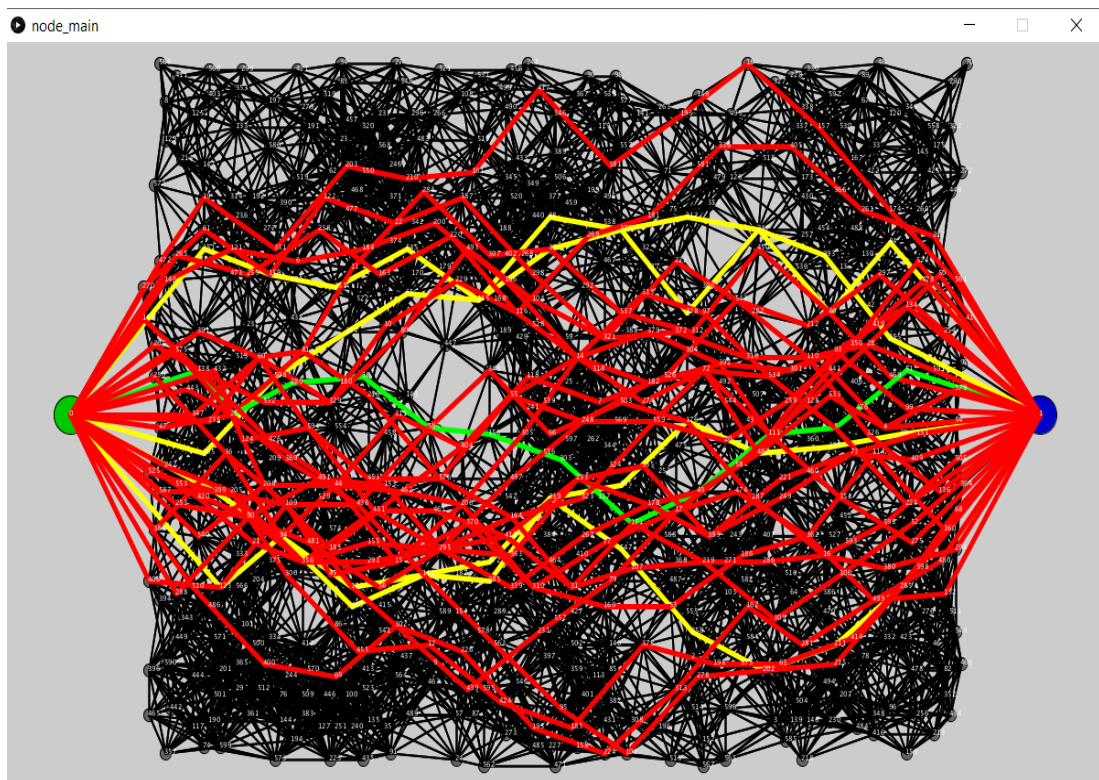


Figure 4.15 Paths for node size of 700

Note: Meaning of each path formed is revised below

1. Green Path - Shortest path considering path distance (this is the Euclidean distance).
2. Yellow Path - Shortest path considering path length, i.e. the number of nodes on the path (minimum number of nodes or hops in the path).
3. Blue Path - Paths that satisfy both the above conditions (has the least number of nodes and is shortest in Euclidean distance too). This occurs rarely.
4. Red Path - All the other paths that do not fit in the above categories.

CHAPTER 5

HARDWARE IMPLEMENTATION

In this chapter detailed explanation of design and implementation of hardware and its working is given. The hardware is designed and implemented for rural applications. The concept of Internet of Things (IoT) is used in the applications. Before going into the details of applications and hardware working, a brief introduction of hardware components and protocols used is illustrated.

5.1 Hardware Components

In this section a brief explanation of hardware components used in each node is given. Each sensor node in the barrier is designed with a temperature sensor so as to send the temperature data from one end of the node to other end and also to upload the temperature data to IoT cloud. One such node of barrier is shown in figure 5.1 given below

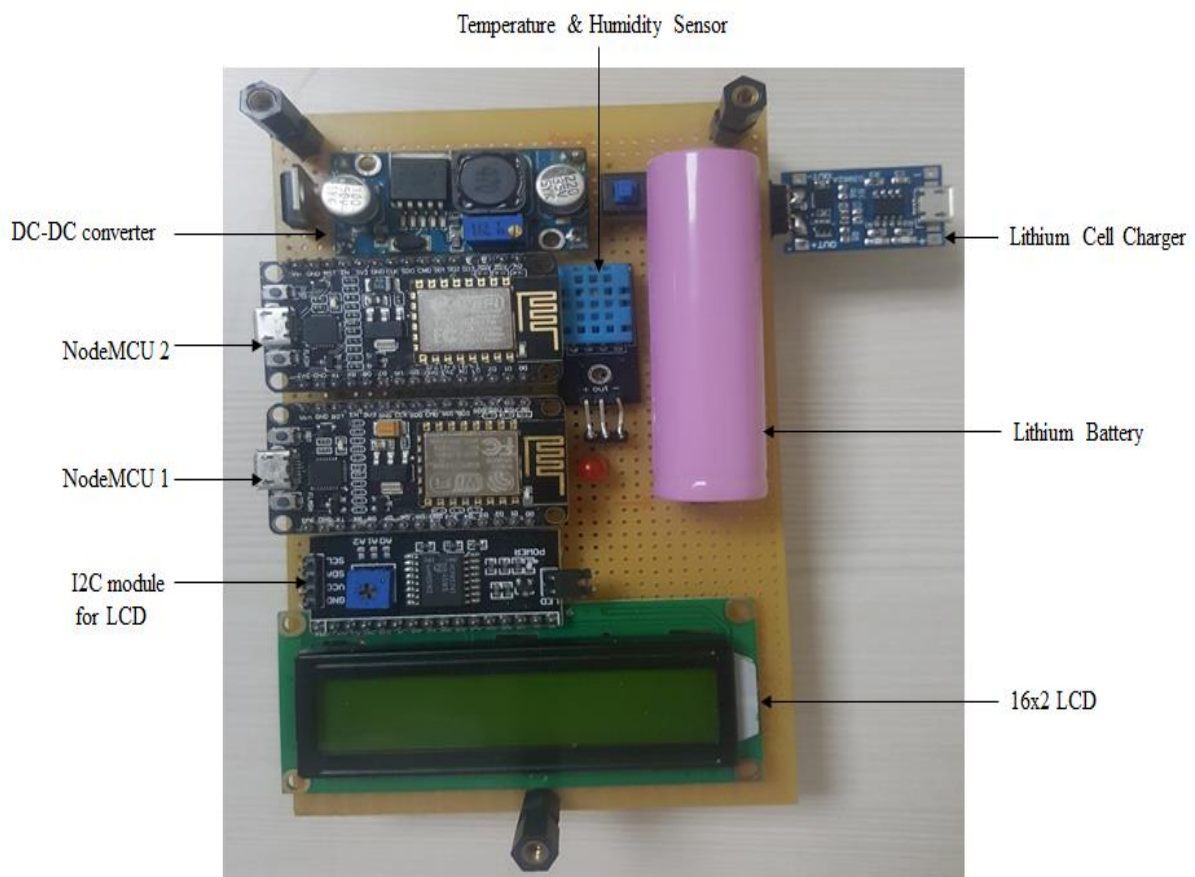


Figure 5.1 Sensor node of a barrier

Following is the list of hardware components with a brief explanation of each

1. NodeMCU: This device provides internet and IoT feature to the sensor node and is considered as heart of the node. Two NodeMCUs are used in each sensor node, one for providing internet and the other to which temperature sensor is connected and also it provides the IoT feature to the node
2. Lithium Battery: As every node is a wireless node, it needs a power source to operate continuously. Lithium battery provides the required power to a node. The lithium battery used in every node is of 12V and 2600mAH capacity.
3. Lithium Cell Charger: The lithium battery used in every node is rechargeable and therefore a charger is included with each node which has a micro-usb port just as our cellphones. As a result, nodes can be charged with normal phone chargers also.
4. 16x2 LCD: Liquid Crystal Display (LCD) is used to display the temperature and humidity data from temperature sensor, the internet connectivity status of node, the service provider of internet and the battery status.
5. I2C module: I2C module is used to provide compatibility between LCD and NodeMCU. LCD has a 8-bit data port which means 8 digital pins of NodeMCU must be connected to LCD which is a disadvantage as 8 pins of NodeMCU will be always busy and cannot be used for any other purpose. Therefore to operate LCD in 4-bit mode I2C is used so that only 4 digital pins of NodeMCU is used and not 8.
6. DC-DC converter: The output of lithium battery is 4V but the NodeMCU and LCD needs 5V, therefore a DC-DC boost converter is used that converts 4V to 7V.
7. Temperature and Humidity Sensor: DHT11 is a temperature and humidity sensor which provides true temperature and humidity values from the environment.

5.2 Hardware for Applications of Barrier Coverage

As stated earlier in this chapter, the applications of barrier coverage are presented with respect to rural areas. Following two cases of rural applications are designed which can be readily applied to rural areas

Case 1: When internet is present in the rural areas, then the barrier coverage can be formed to

1. Provide Wi-Fi range extension from the nearest place to the rural area through barrier of nodes
2. Forest Fire detection through IoT.

Case 2: When internet is not present

1. Forest Fire detection through multi-hop data transfer.

Before explaining each application in detail, a detailed explanation of operating modes of NodeMCU is required as these modes will decide the application.

5.2.1 Modes of NodeMCU: NodeMCU also called as ESP8266 is a microcontroller that comes with an in-built Wi-Fi shield that gives a provision for the user to connect ESP8266 to internet. NodeMCU gets connected to the internet just as mobile phone does with a username and password and works in two modes, which are

1. **Station (STA) mode**
2. **Access point (AP) mode**

Devices which get connected to available Wi-Fi networks are called as Stations and the devices that provide the Wi-Fi network are called as Access Points, example routers.

1. Station (STA) mode

In STA mode, ESP8266 acts as an end station just like the mobile phone, when it gets connected to a Wi-Fi network. To connect to the internet, NodeMCU must be provided with an access point. These access points in turn get connected to internet service provider through a wired connection or wireless connection. A wired access point can be a router and a wireless access point can be a mobile phone hotspot. ESP8266 in STA mode can connect to router or mobile phone hotspot. Every access point will have a unique identifier called as Service Set Identifier (SSID) and a password. Figure 5.2 shows the operation of NodeMCU in STA mode.

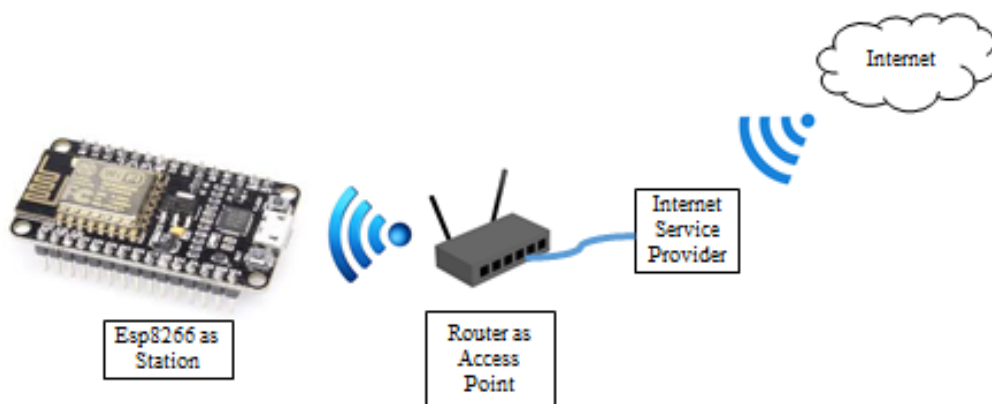


Figure 5.2 NodeMCU in STA mode

Example code:

```
#include<ESP8266WiFi.h>

const char* ssid="access points name";
const char* password="password";

void setup()
{
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while(WiFi.status()!=WL_CONNECTED) // to get status of wifi connection
  {
    delay(1000);
    Serial.print(".");
    Serial.print("connected");
    Serial.println(WiFi.localIP()); // to get IP address of NODEMCU
    Serial.println(WiFi.macAddress());
  }
}

void loop()
{
  Serial.println(WiFi.RSSI()); // to get signal strength of WiFi
  delay(1000);
}
```

Explanation of the code:

Connecting ESP8266 to a Wi-Fi network will start with defining a simple begin function and the parameters that will be passed to this function are SSID and password. Maximum length of SSID is 32 characters and minimum length of password is 8 characters and maximum length should not exceed 64 characters. The simplest form of begin function is depicted as

WiFi.begin (ssid, password);

Apart from the SSID and password, which are the compulsory parameters to be given, there are some optional parameters also as depicted in the function below

WiFi.begin (ssid, password, channel, bssid, connect);

Where:

- channel parameter selects a specific channel
- bssid is the parameter that returns the Medium Access Control (MAC) address of access point.
- connect parameter, which will by default be true but if set to false will save all the other parameters without actually connecting the ESP8226 module to internet.

`#include<ESP8266WiFi.h>` is the WiFi library, which essentially consists of code that is needed to connect to the network. The second and third statements provide the necessary credentials required to connect to the network i.e., name of access point and its password. In void setup, initially the baud rate of serial terminal, which is used to display the output, is defined. The `WiFi.begin` function is passed with the parameters given in the second and third statement. The while loop continuously checks whether the ESP module is being connected to internet or not and till the time the connection is established, a dot, every second, will be displayed on serial terminal. Once the connection is established, a “connected” message appears on the serial terminal.

The function `serial.println (WiFi.localIP ())`; returns the IP assigned to ESP module by the access point and the function `Serial.println (WiFi.macAddress ())`; returns the MAC address of ESP. The void loop function continuously returns the strength of the connection of WiFi service provider with a delay of one second between each return value. Here RSSI stands for received signal strength indicator, which defines the signal strength of the access point providing internet connection to the NodeMCU. The RSSI value will in decibel. The return value will always be negative as it is the logarithmic value of RSSI. If the RSSI value is near -40dB, then it is an indication that the signal strength is at its best and as the value goes beyond -40dB then it means that signal strength is reducing.

Output of Example code:

As shown in the output, line 1 displays the successful connection message. Line 2 returns the IP assigned to ESP board by the access point. Line 3 is the MAC address of ESP8266 and from line 4, the value of RSSI is shown. If the WiFi connection is lost, ESP will automatically reconnect to the last network once the connection is up.

```
COM3
Send
.connected
192.168.43.233
84:F3:EB:B7:C2:31
-43
-42
-42
-42
-38
-46
-49
-50
-44
-41
-44
-44
-43
```

Autoscroll No line ending 9600 baud Clear output

2. Access point (AP) mode

As stated earlier an access point is a device that provides wireless internet connectivity to the other devices but it itself gets internet through a wired connection. ESP8266 can do the same function but with only one difference, here ESP8266 gets internet connection wirelessly. When ESP8266 is operated in such mode then it is called as Access Point (AP) mode or soft-Access Point mode where it works as a hot-spot for providing internet. There is a maximum limit in the number of stations that can be connected to ESP8266 AP. At one time not more than five stations can be connected. The basic set-up of ESP8266 working in AP mode is depicted in figure 5.3.

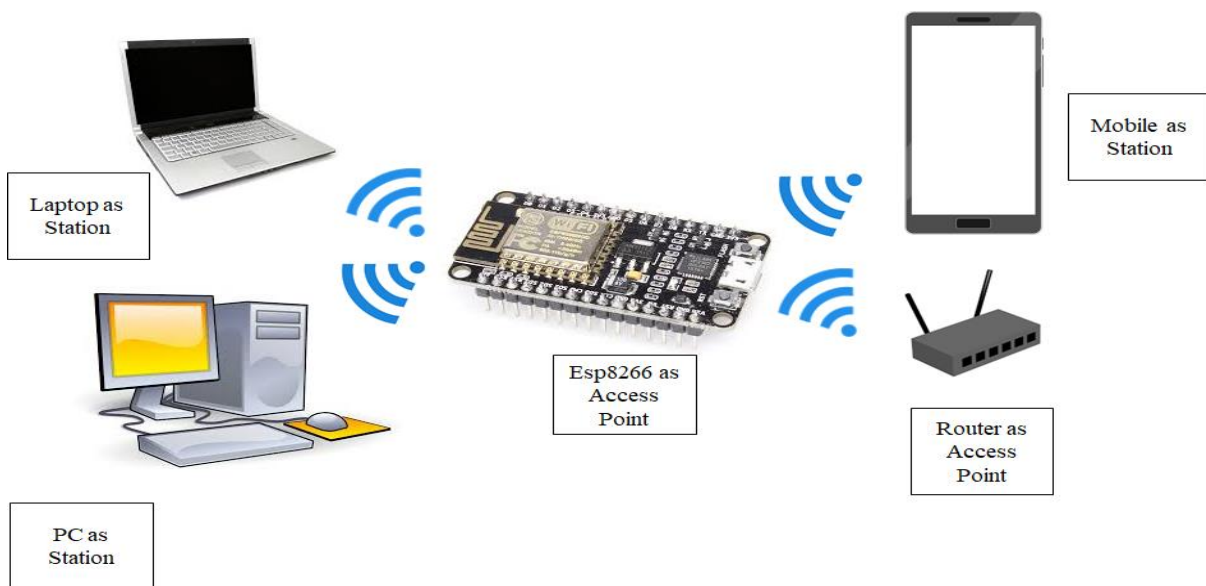


Figure 5.3 ESP8266 in AP mode

As shown in figure 5.3 laptop and PC were not in the range of router and therefore could not connect to internet provided by the router. In such case, first ESP8266 is connected to the router and then operated in AP mode after which PC and laptop will be connected to ESP8266.

5.2.2 Connecting ESP to one out of multiple access points: If the current Wi-Fi is unavailable then there should be an option for NodeMCU to connect to the next available Wi-Fi. For this, ESP8266WiFiMulti library is used. To make sure that NodeMCU module connects to one out of many Wi-Fi networks names of access points should be added. Once added, ESP8266 will connect to an access point which will have the strongest signal. Method of adding access points is depicted below:

```
wifiMulti.addAP ("access point 1", "password");  
wifiMulti.addAP ("access point 2", " password");  
wifiMulti.addAP ("access point 3", " password");
```

5.2.3 Dynamic Configuration: Most of the routers will allocate an IP (Internet Protocol) address to devices dynamically. That means a device gets a new IP each time it connects to the internet provided by the router. ESP8266 in station mode when gets connected to an access point will get IP dynamically. But what if there is a static internet connection available. In this case before running the begin function, the configuration function in ESP8266 has to be executed. In configuration function, a local IP, which is static, will be passed as a parameter. One such function is given below

```
WiFi.config (local_ip, gateway, subnet, dns1, dns2);
```

Where:

- local_ip is the IP to be assigned to the ESP
- gateway is the gateway address (access point IP) which is used to access the external network
- subnet gives the range of IPs
- dns1 and dns2 are optional

5.3 Wi-Fi Range Extender for Rural Applications through Barrier Coverage

One of the applications of barrier coverage as stated in this thesis is Wi-Fi range extender. For this more than one NodeMCU is required and the internet from the source point can be

extended to a distance where sink point is available. Set-up of one such application is given in figure 5.4



Figure 5.4 Set-up for Wi-Fi range extender

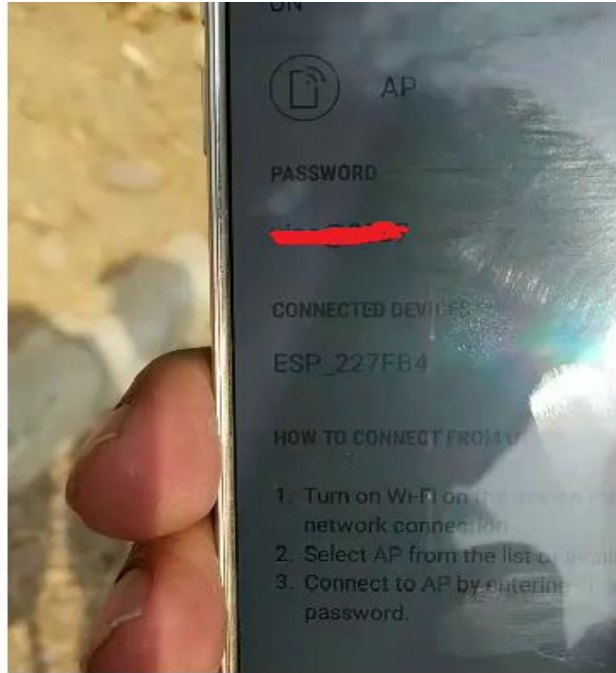
In this set-up, the first NodeMCU gets connected to the main access point which is a router and will then act as a hot-spot providing internet to second NodeMCU. Now the second ESP8266 will connect to the internet provided by first and will again act as hot-spot. In this way the internet of router or mobile phone can be extended to a distant place, which otherwise is not possible without the use of NodeMCU.

5.3.1 Experiment with 3 nodes: In this section the outcome of a real-time experiment conducted with 3 nodes is shown which will prove and validate that NodeMCU can be used as Wi-Fi extender. Figure 5.5 (a), (c) and (d) consists of 3 sensor nodes, each with NodeMCU used in STA mode to connect to internet and in AP mode to provide hot-spot to other devices. Figure 5.5 (d) is the mobile used as main hotspot. Following outcomes were recorded from this experiment:

1. Figure 5.5 (a) is first node which is connected to mobile hot-spot named as AP.
2. This node is acting as hot-spot by name AP1.
3. Figure 5.5 (b) is the mobile whose hot-spot name is AP and under connected devices list ESP is seen. This proves that first nodeMCU is connected to mobile hotspot.
4. Figure 5.5 (c) is second nodeMCU named as AP2 and connected to AP1.
5. Figure 5.5 (d) is third nodeMCU named as AP3 and connected to AP2.



(a) Node 1 connected to mobile hotspot



(b) Mobile hotspot



(c) Node 2 connected to Node 1



(d) Node 3 connected to Node 2

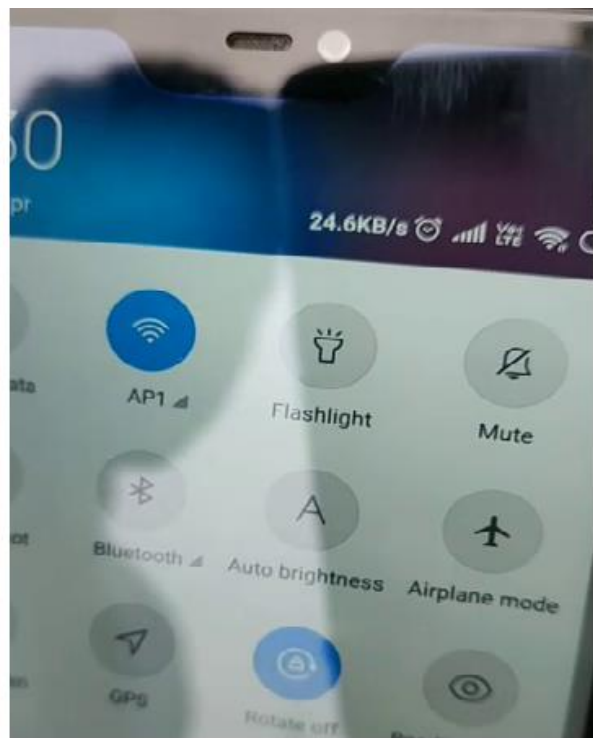
Figure 5.5 Wi-Fi Extender experiment with 3 nodes

5.3.2 Speed test in Wi-Fi range extender: In section 5.3.1, range extension test was successfully implemented with 3 nodes. In this section a speed test is conducted for those 3 nodes by connecting a mobile phone to the hot-spots provided by the 3 nodeMCUs. Following outcomes were recorded from this experiment:

1. In figure 5.6 (a), the mobile is connected to main hotspot which is the mobile phone used in figure 5.5 (b). In every mobile a High-Definition (HD) video is run at the background starting from main hot-spot
2. In figure 5.6 (a), the mobile is connected to AP and the internet speed shown is 56.7 KB/s which is 453 kbps.
3. In figure 5.6 (b), the mobile is connected to node 1 which is AP1 of figure 5.5 (a). The speed here is 24.6KB/s which is 197 kbps. Here the reason for reduced speed is obvious due to the fact that AP1 is receiving internet from AP.
4. Mobile in figure 5.5 (c) is connected to AP2 which is receiving internet from AP1 and therefore here the internet speed is further reduced and is 8.3 KB/s which is 66 kbps.
5. At last mobile of figure 5.5 (d) is connected to AP2 which is receiving internet from AP2 and therefore the speed is further reduced to 0.3 KB/s which is 2.4 kbps.



(a) Mobile connected to hotspot



(b) Mobile connected to Node 1

Conclusion from the above experiments is that NodeMCU can be used to extend the range of Wi-Fi without any limit but the speed of internet reduces proportional to distance. Point to be noted here is that, speed will reduce if and only if the nodes nearer to the main hot-spot are using the internet.



(c) Mobile connected to Node 2

(d) Mobile connected to Node 3

Figure 5.6 Speed Test of Wi-Fi Range Extender

5.4 IoT based Forest Fire Detection System for Rural Applications

In this section, an application of barrier paths in designing forest fire detection and monitoring system is proposed using NodeMCU and open source IoT cloud called as Message Queuing Telemetry Transport (MQTT). Before putting forward the details of the design, a detailed explanation of MQTT is given below.

5.4.1 Introduction to MQTT: MQTT is a light-weight, in terms of overhead required and power consumption, IoT protocol which is used for establishing client-to-client communication through MQTT server. It is an open IoT source protocol used in Machine-to-Machine (M2M) in situations where the need is of small source code and when availability of communication bandwidth is on lower side. As MQTT is a lightweight protocol, the power consumption is very low and therefore is best suitable to be used in situation where battery requirement is minimum. Figure 5.7 given below describes the architecture of MQTT. Architecture of MQTT is based on Publish-Subscribe model, where publish is synonymous to transmit and subscribe to receive.

Virtual communication channels called as ‘topic’ has to be created before transmitting or receiving data through MQTT and the data to transmitted or received will be published or subscribed, respectively, to this topic. For example a temperature sensor, as shown in figure 5.7 wants data to be uploaded temperature values to MQTT, then it has to publish data to a topic say ‘temp’, and at the other end a client, say a mobile phone wants to view the temperature values then it has to subscribe to topic ‘temp’. A third party gateway (can also be called as IoT gateway) that handles the connection between the clients is called as **Broker**.

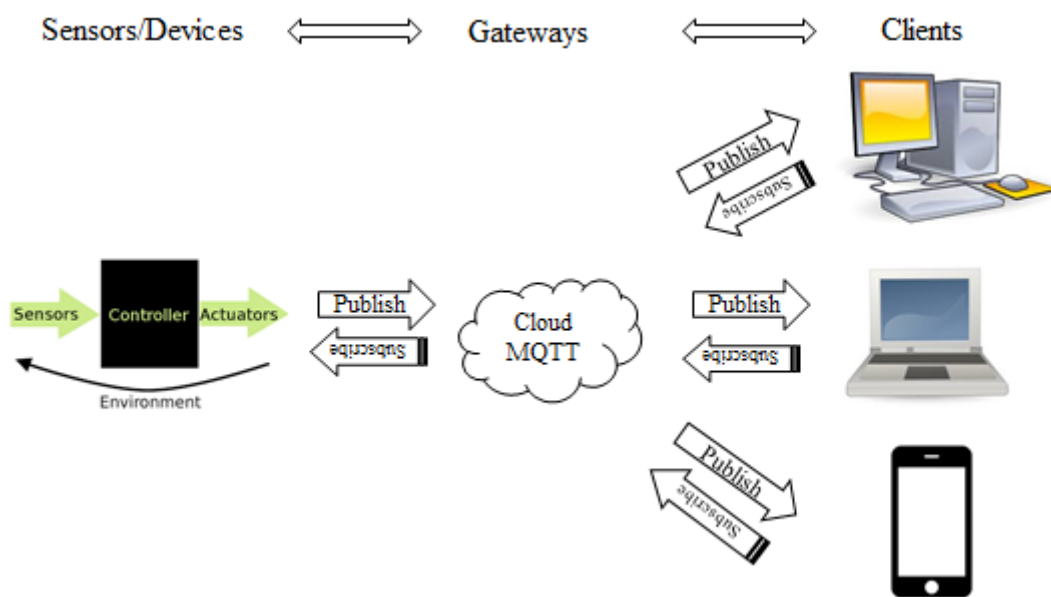


Figure 5.7 Architecture of MQTT

The work of MQTT broker is to channel the incoming messages, filter them and convey correctly to the intended clients.

5.4.2 Use of MQTT in forest fire detection: In this experiment of forest fire detection, MQTT is used as server and software by name mqtt box will be used as broker in PC/Laptop and an app named as MQTT dashboard will be used as broker in mobile phone. Therefore, with the use of MQTT server and MQTT broker the temperature and humidity data of DHT11 sensor which is connected to NodeMCU can be viewed on PC/Laptop, mobile from any part of the world.

5.4.3 Connecting NodeMCU with MQTT: To use MQTT cloud one should have the software named MQTT box in Laptop/PC. For mobile platforms, app by name MQTT

dashboard is available in android play store and MQTT buddy in iOS. The stepwise procedure for connecting MQTT box with cloud MQTT is given below

1. Open www.cloudmqtt.com , sign-up and create a profile
2. Create a instance and open it
3. Install MQTT box in PC/Laptop and download the MQTT dashboard app in the mobile
4. In MQTT box click on create MQTT client and enter the client name.
5. Select protocol as mqtt/tcp, host name should be server name followed by : and port number (Refer the figure 5.8)
6. Save the current configuration of MQTT box and as soon as save option is clicked, the connection tab should turn green (figure 5. 9) which is an indication of successful connection establishment.

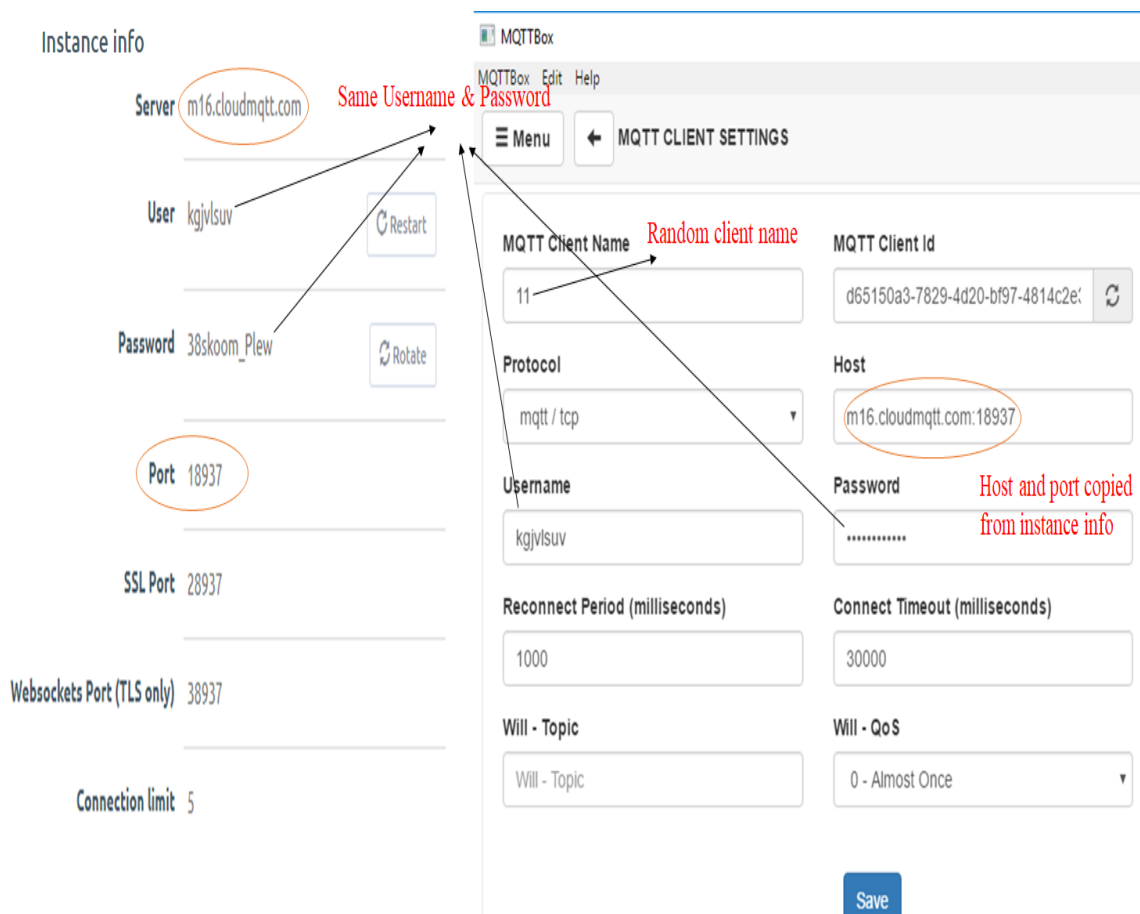


Figure 5.8 Configuration between MQTT box and cloudmqtt.com

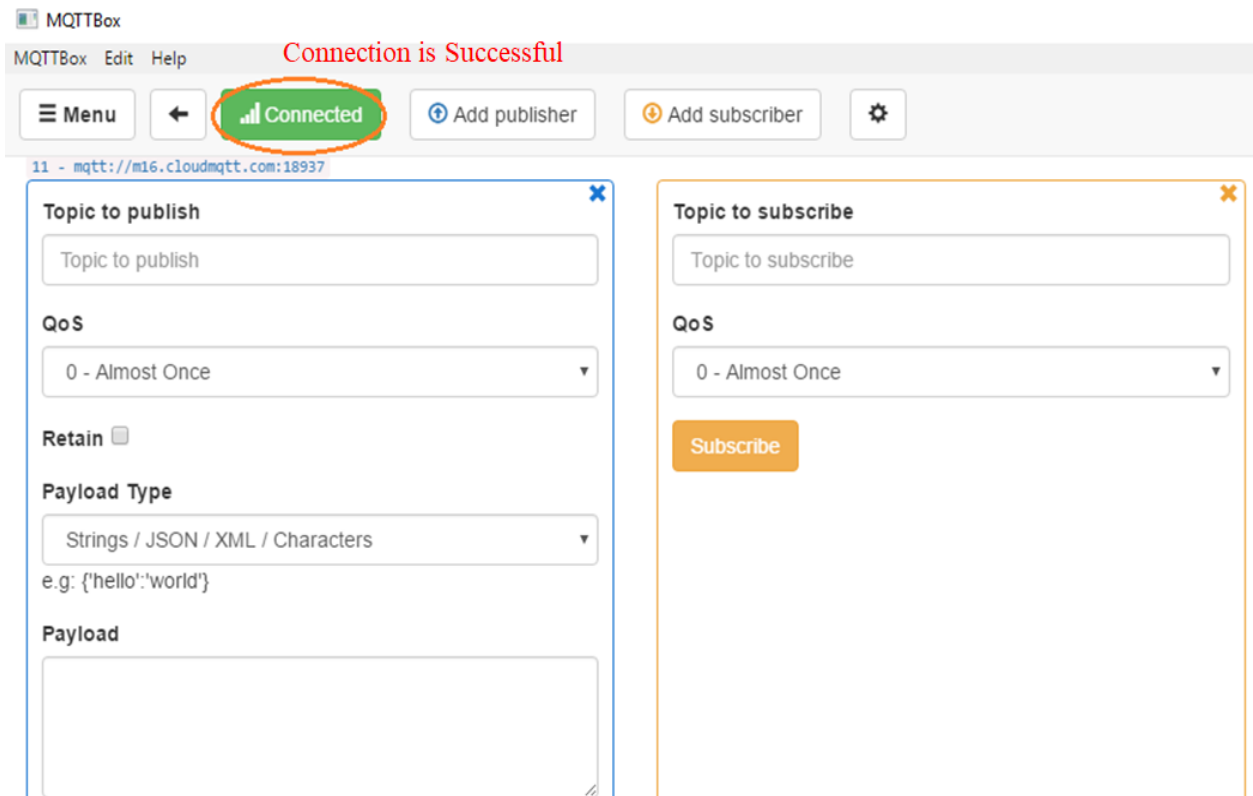
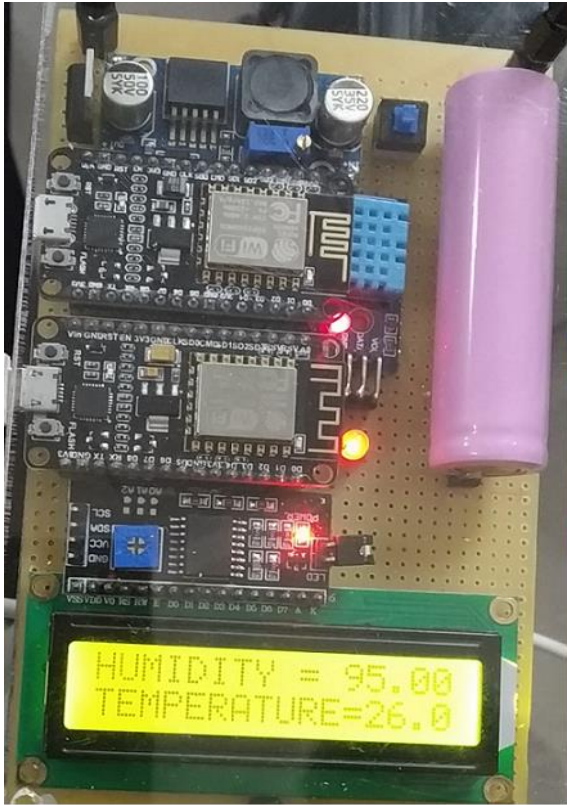


Figure 5.9 Successful connection between MQTT box and cloudmqtt.com

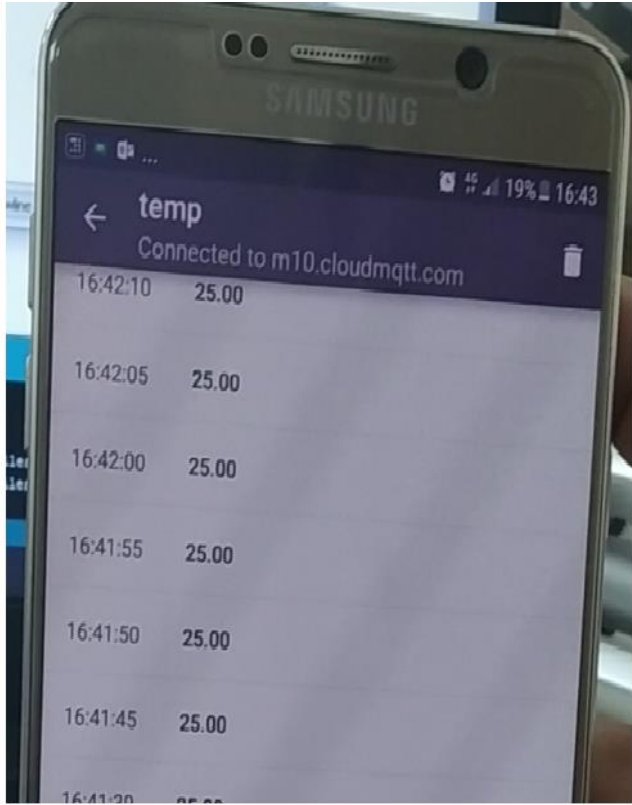
5.4.4 Experiment for uploading DHT11 data to IoT through MQTT: In this experiment, DHT11 sensor is interfaced with NodeMCU and the temperature and humidity data is uploaded to MQTT cloud and the same is viewed on laptop and mobile.

Following outcomes were observed during the experiment

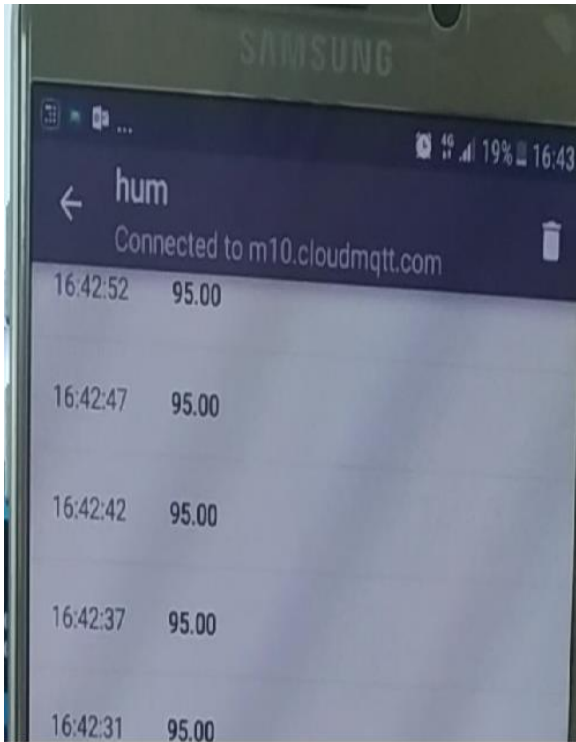
1. Figure 5.10 (a) represents the sensor node with DHT11 sensor interfaced to NodeMCU and on the LCD temperature and humidity data is shown. In this node a code is written to upload temperature and humidity data to MQTT cloud. The topics used are 'temp' for temperature and 'hum' for humidity.
2. Figure 5.10 (b) is of MQTT box in mobile 1 which is connected to cloudmqtt.com and is subscribed to topic 'temp' and therefore is receiving the temperature data from node continuously along with time stamp.
3. Figure 5.10 (c) is of MQTT box in mobile 1 which is connected to cloudmqtt.com and is subscribed to topic 'hum' and therefore is receiving the humidity data from node along with time stamp.
4. Figure 5.10 (d) is of MQTT box in mobile 2 which is connected to cloudmqtt.com and is subscribed to topic 'temp' and therefore is receiving the temperature data from node along with time stamp.



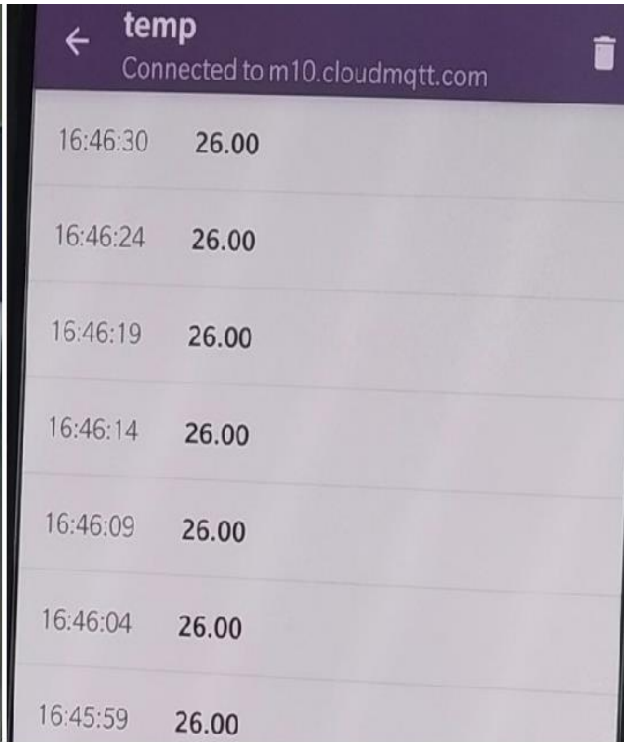
(a) Node for Forest Fire Detection



(b) Temperature data on mobile 1



(c) Humidity data on mobile 1



(d) Temperature data on mobile 2

Figure 5.10 Uploading DHT11 data to IoT Cloud

5.5 Forest Fire Detection through Multi-hop Data Transfer

In this section, forest fire detection and monitoring is implemented without the help of internet and IoT feature. When internet is not present, data can be hopped from source to sink using multiple nodes. Here Xbee, which is a Radio Frequency (RF) transceiver, is used to transfer the data from one end of the barrier to the other end through multiple hops. An experiment with four nodes was conducted where carbon oxide (CO) sensor was attached to one node fitted on drone along with Xbee transceiver. The data from this node was sent to the second node which was having only Xbee transceiver. Data from second node was sent to third and from third to fourth. Consider the figure 5.11 which shows the setup of 3 drones and a base station. In this data from drone 1 will go to the base station through intermediate drones.

5.5.1 Experiment for implementing forest fire detection through multi-hop data transfer: In this experiment, forest fire detection is implemented when internet is not present and sensor data is sent to nearest base station. For this, two drones are deployed in the open area as given in figure 5.12, out of which only one drone will have a gas sensor fitted on it. From this drone the CO value is transmitted to the second drone from where the sensor value reaches the base station. Figure 5.12 (a) show the deployment of drones and 5.12 (b) is showing the data received at base station.

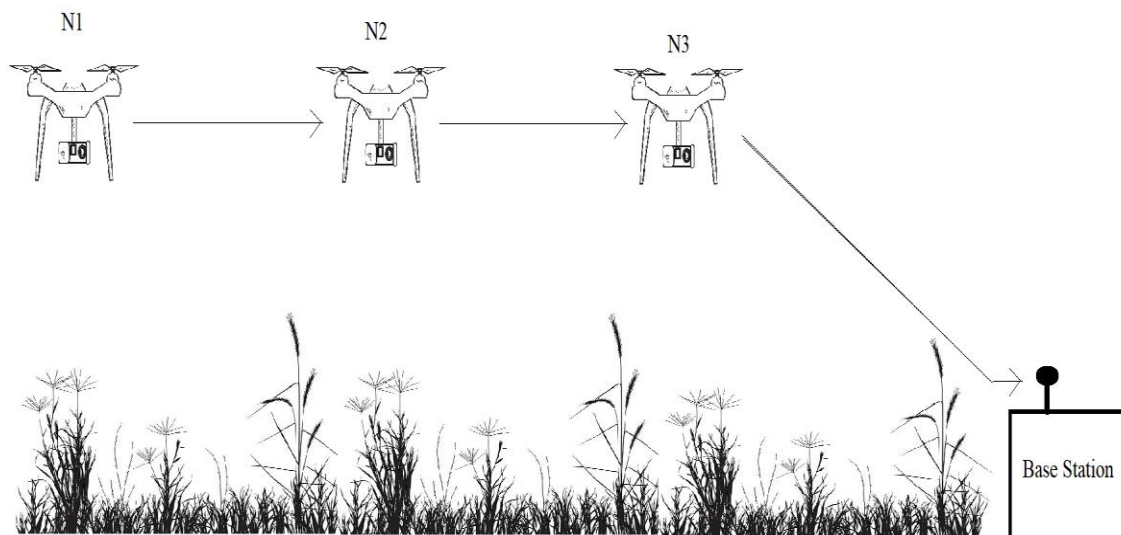


Figure 5.11 Set up of sensor nodes for forest fire detection



(a) Drones deployed



(b) Data received at base station

Figure 5.12 Experiment set up for multi-hop data transfer

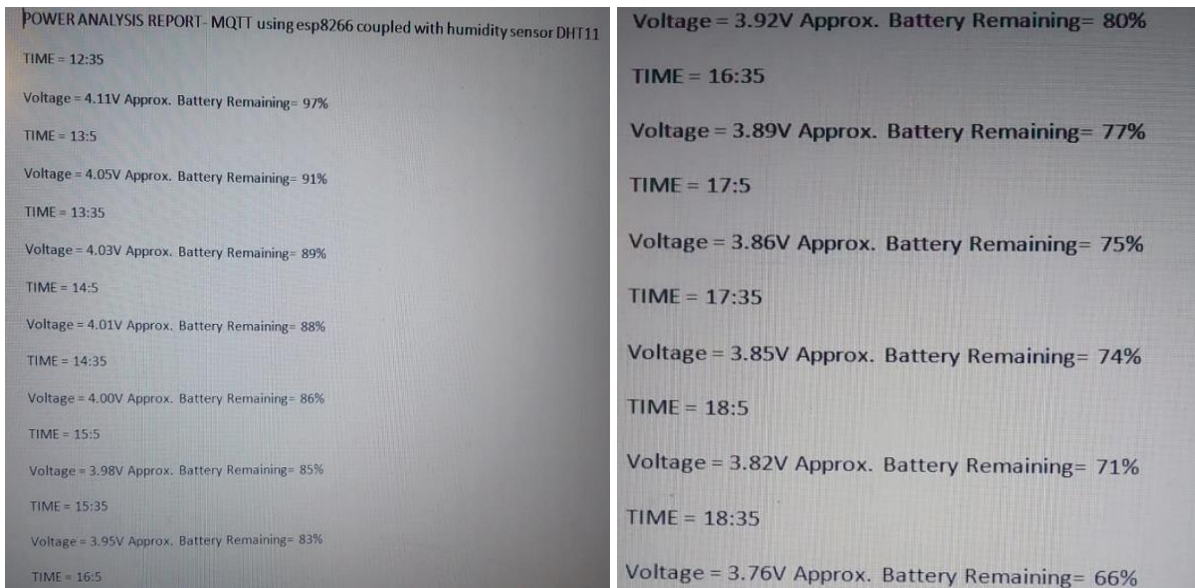
5.6 Power Consumption Analysis of Sensor Node

As each sensor node is battery operated, the operating lifetime is limited and is crucial for barrier coverage. In this section, power consumption analysis of sensor node is performed. As stated earlier, each node is fitted with two NodeMCUs, DC-DC power booster, an I2C module and a LCD. Each of these devices consumes power from the standalone lithium battery whose capacity is 2600mAh. Therefore it becomes essential to compute and analyse the operating lifetime of each node when all these devices are powered on and operated continuously.

To perform the power analysis two cases are considered,

1. **Case 1:** When the sensor nodes are working for Wi-Fi range extension
2. **Case 2:** When sensor nodes are continuously uploading of sensor data to MQTT cloud

In the first case, the node was fitted with DHT11 sensor from which values of temperature and humidity were uploaded to MQTT cloud. The result of the experiment conducted for power analysis is given in figure 5.13 (a) to (c). The experiment was started at 12:35pm when the battery was 100%. For every 30 minutes interval, the remaining battery percentage was recorded and displayed on the serial terminal of the NodeMCU. For the first 7 hours, the drop in battery capacity was sequential and in 7 hours only 45% of the battery was consumed. But after 7 hours there was a sudden drop observed within 30 minutes the complete battery was dead.



(a) Power analysis from 100% to 80%

(b) Power analysis from 80% to 60%



(c) Power analysis from 60% to 0%

Figure 5.13 Power consumption analysis of sensor node

One reason for this is due to the fact that when the test was started, the voltage supplied by the NodeMCU and other devices need a minimum of 3.5V and when voltage reaches 3.5V or less then these devices cease to operate, as a result of which there was no output from the system. The hardware testing has shown that practical application of barrier coverage for rural application can be implemented in a easy and hassle-free manner.

CHAPTER 6

CONCLUSION

In this research design and implementation of k-barrier coverage for rural applications is implemented with the use of Minimum Non-Overlap Radius Deployment Algorithm (MNORDA). MNORDA is a pre-deployment processing algorithm which helps in maintaining minimum overlap radius between two nodes. As a result of which no two nodes will be deployed within the minimum radius which guarantees uniform coverage of the network. The extensive literature survey is carried out in order to gauge the existing state of the technology developed and finding the research gap in this area.

The literature review yielded the following gaps in the existing research

1. Message overhead required in centralized approach hampers the overall operating lifetime of the network when compared to distributed approach.
2. Line-based and curve-based deployment techniques do not provide full coverage of the network.
3. Directional sensing approach uses sensors capable of sensing only in one direction and therefore has limited coverage ability.
4. Mobile sensors for barrier coverage formation needs overhead in terms of message and power required for moving the sensors.

Based on the above gaps the objectives were defined and achieved as below

1. Successful deployment of the nodes using minimum non-overlap radius (O_R) uniformly and evenly throughout the network for random deployment. The node size tested 10, 20, 100, 200, 300, 400, 500, 700 and 900 nodes.
2. Successful construction the barrier paths from source node on the left side of the region to sink node on the right side of the region. Also the proposed MNORDA algorithm was successful in maximizing the number of barrier formed for all node sizes when compared to already existing works
3. Once the barrier paths are formed, maintaining the barrier for longer operating time is one of the prime requirement. Therefore the lifetime of each barrier should be as high as possible. MNORDA algorithm has achieved a considerably more lifetime when compared other state of art works.

4. Once the complete network has been formed, providing information of every node to every other node helps in discovering the complete network. For this adjacency matrix for every node is constructed, entries of which represents the information of neighboring node the current node has.
5. Adjacency matrix helps in identifying the isolated nodes in the network and switching off these isolated nodes will reduce the overall power consumption of the network and thereby increasing lifetime. To identify an isolated node, the adjacency matrix of that node will have all 0's in its rows and columns.
6. MNORDA algorithm identifies the shortest paths in the network in terms of distance between source and sink nodes and also in terms of number of nodes in each path.

The methodology adopted to achieve each objective so defined is explained in detailed manner in the subsequent chapters. The MNORDA algorithm developed to achieve the objectives have functions associated with it which are called and executed whenever required. Example networks are considered where ever necessary. The process of generating adjacency matrix is also explained followed by the novel technique of minimizing the packets exchanged in the network. Construction of barrier paths followed by the technique of identifying the shortest path in terms of number of nodes and distance has been presented.

Simulation results of MNORDA in comparison with the other existing state-of-art works were performed. MNORDA has outperformed its counterparts by a good margin for the following parameters.

1. **Computing maximum number of barriers formed for different node size:** Number of barriers formed was almost 3 times more showing an improvement of 300% for node size of 100 when compared to other algorithms. For node size of 200, 300, 400 and 500, MNORDA outperformed its counterpart algorithms AHVGB, MobiBar and DDABC by a great margin and the only algorithm which was close to MNORDA is CBCDLA.
2. **Power consumption in terms of number of packets exchanged between nodes for different node size:** Here as number of barriers formed in MNORDA therefore the number of packets exchanged were also more than other algorithms. Given a trade-off between number of barriers and number of packets exchanged, it is guaranteed that MNORDA will perform better.

3. **Termination time of the algorithm for different node size:** MNORDA was the fastest algorithm to converge in finite time for node size of 100, 200, 300 and 400. Minimum time taken by the algorithm to converge was 0.5 seconds for network of 100 nodes and maximum time taken was 37 seconds for 400 nodes. No other algorithm was close to the performance of MNORDA.
4. **Lifetime estimation of barriers for different node size:** Lifetime of the network is estimated in terms of number of weeks. The lifetime obtained by MNORDA ranges from 20 weeks for 100 nodes to 120 weeks for 500 nodes
5. **Sensor Utilization for different node size:** To calculate number of sensors utilized in forming barriers to the number of sensors deployed, number of sensors in each path are computed and then added to obtain sensor utilization. MNORDA outperforms MDP, BCA and TOBA algorithms by a minimum of 20% to maximum of 50% for 300 nodes and by minimum of 4% to maximum of 34% for 600 nodes.

6.1 Highlights of proposed MNORDA algorithm

The proposed algorithm covers the gaps provided by literature survey by offering some novel features. This algorithm is a pre-deployment processing algorithm where sensors once deployed need not be moved. All the sensors have Omni-Directional communication capability that guarantees formation of maximum barrier paths for the given number of sensors while providing information of every node in the network to every other node. MNORDA algorithm can work for both homogeneous and heterogeneous sensing range. Identification of isolated nodes allows enhancing the lifetime of network by switching off these nodes permanently.

6.2 Future Scope

In future scope, research can be extended to try to minimize the number of packets exchanged between the nodes which help in enhancing the operating lifetime of network. Research can be further extended to propose a trade-off value between number of barrier formed and number of packets exchanged so that this trade-off defines optimal value in the network.

At the level of the research can be extended to work on design and implementation of forest fire fighting robots by forming barrier paths in the fire affected area. One more open design problem is to implement consensus technique with the use of drones in finding shortest escape path in case of fire as the conflagration of forest fire is still a serious concern.

REFERENCES:

1. Gage, D.W., *Command control for many-robot systems*, 1992, Naval Command Control and Ocean Surveillance Center Rdt And E Div San Diego CA.
2. Kumar, S., T.H. Lai, and A. Arora. *Barrier coverage with wireless sensors*. in *Proceedings of the 11th annual international conference on Mobile computing and networking*. 2005. ACM.
3. Han, R., W. Yang, and L. Zhang, *Achieving Crossed Strong Barrier Coverage in Wireless Sensor Network*. *Sensors*, 2018. **18**(2): p. 534.
4. Chen, A., S. Kumar, and T.H. Lai, *Local barrier coverage in wireless sensor networks*. *IEEE Transactions on Mobile Computing*, 2010. **9**(4): p. 491-504.
5. Nguyen, T.G. and C. So-In, *Distributed deployment algorithm for barrier coverage in mobile sensor networks*. *IEEE Access*, 2018. **6**: p. 21042-21052.
6. Saipulla, A., et al., *Barrier coverage with line-based deployed mobile sensors*. *Ad Hoc Networks*, 2013. **11**(4): p. 1381-1391.
7. Fan, F., et al., *Dynamic Barrier Coverage in a Wireless Sensor Network for Smart Grids*. *Sensors*, 2019. **19**(1): p. 41.
8. Saipulla, A., et al. *Barrier coverage with sensors of limited mobility*. in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. 2010. ACM.
9. Kong, L., et al., *Adaptive barrier coverage using software defined sensor networks*. *IEEE Sensors Journal*, 2016. **16**(20): p. 7364-7372.
10. Wang, Z., et al., *Cost-effective barrier coverage formation in heterogeneous wireless sensor networks*. *Ad Hoc Networks*, 2017. **64**: p. 65-79.
11. Wang, Z., et al., *Achieving location error tolerant barrier coverage for wireless sensor networks*. *Computer Networks*, 2017. **112**: p. 314-328.
12. Xu, B., et al., *Strengthening barrier-coverage of static sensor network with mobile sensor nodes*. *Wireless Networks*, 2016. **22**(1): p. 1-10.
13. Rout, M. and R. Roy, *Self-deployment of randomly scattered mobile sensors to achieve barrier coverage*. *IEEE Sensors Journal*, 2016. **16**(18): p. 6819-6820.
14. Cobb, J.A. *Improving the lifetime of non-penetrable barrier coverage in sensor networks*. in *2015 IEEE 35th International Conference on Distributed Computing Systems Workshops*. 2015. IEEE.
15. Kim, D., et al., *Maximum lifetime dependable barrier-coverage in wireless sensor networks*. *Ad Hoc Networks*, 2016. **36**: p. 296-307.
16. Saipulla, A., et al. *Barrier coverage of line-based deployed wireless sensor networks*. in *IEEE INFOCOM 2009*. 2009. IEEE.

17. Du, J., et al., *Maximizing the lifetime of k -discrete barrier coverage using mobile sensors*. IEEE Sensors Journal, 2013. **13**(12): p. 4690-4701.
18. Chen, D.Z., et al., *Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain*. Discrete & Computational Geometry, 2013. **50**(2): p. 374-408.
19. Li, S. and H. Shen. *Minimizing the maximum sensor movement for barrier coverage in the plane*. in *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2015. IEEE.
20. Shen, C., et al. *Barrier coverage with mobile sensors*. in *2008 International Symposium on Parallel Architectures, Algorithms, and Networks (i-span 2008)*. 2008. IEEE.
21. Nguyen, T.G., C. So-In, and N.G. Nguyen, *Barrier coverage deployment algorithms for mobile sensor networks.*, 2017. **18**(7): p. 1689-1699.
22. Bhattacharya, B., et al., *Optimal movement of mobile sensors for barrier coverage of a planar region*. Theoretical Computer Science, 2009. **410**(52): p. 5515-5528.
23. Bar-Noy, A., D. Rawitz, and P. Terlecky. "*Green*" *Barrier Coverage with Mobile Sensors*. in *International Conference on Algorithms and Complexity*. 2015. Springer.
24. He, S., et al., *Curve-based deployment for barrier coverage in wireless sensor networks*. IEEE Transactions on Wireless Communications, 2014. **13**(2): p. 724-735.
25. Dobrev, S., et al., *Complexity of barrier coverage with relocatable sensors in the plane*. Theoretical Computer Science, 2015. **579**: p. 64-73.
26. Kong, L., et al. *Automatic barrier coverage formation with mobile sensor networks*. in *2010 IEEE International Conference on Communications*. 2010. IEEE.
27. Jia, J., et al., *An autonomous redeployment algorithm for line barrier coverage of mobile sensor networks*. International Journal of Ad Hoc and Ubiquitous Computing, 2014. **16**(1): p. 58-69.
28. Ban, D., et al., *Energy-efficient algorithms for k -barrier coverage in mobile sensor networks*. International Journal of Computers Communications & Control, 2010. **5**(5): p. 616-624.
29. Wang, Y., et al., *Minimizing mobile sensor movements to form a line K -coverage*. Peer-to-Peer Networking and Applications, 2017. **10**(4): p. 1063-1078.
30. Silvestri, S. and K. Goss, *MobiBar: An autonomous deployment algorithm for barrier coverage with mobile sensors*. Ad Hoc Networks, 2017. **54**: p. 111-129.
31. Cheng, T.M. and A.V. Savkin. *A problem of decentralized self-deployment for mobile sensor networks: Barrier coverage between landmarks*. in *2009 IEEE International Conference on Control and Automation*. 2009. IEEE.
32. He, S., et al. *Cost-effective barrier coverage by mobile sensor networks*. in *2012 Proceedings IEEE INFOCOM*. 2012. IEEE.
33. Kong, L., et al. *Mobile barrier coverage for dynamic objects in wireless sensor networks*. in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*. 2012. IEEE.

34. He, S., et al., *Mobility and intruder prior information improving the barrier coverage of sparse sensor networks*. IEEE Transactions on Mobile Computing, 2014. **13**(6): p. 1268-1282.
35. Tao, D., et al., *Strong barrier coverage in directional sensor networks*. Computer Communications, 2012. **35**(8): p. 895-905.
36. He, J. and H. Shi, *Constructing sensor barriers with minimum cost in wireless sensor networks*. Journal of Parallel and Distributed Computing, 2012. **72**(12): p. 1654-1663.
37. Mostafaei, H., *Stochastic barrier coverage in wireless sensor networks based on distributed learning automata*. Computer Communications, 2015. **55**: p. 51-61.
38. Yang, G. and D. Qiao. *Multi-round sensor deployment for guaranteed barrier coverage*. in *2010 Proceedings IEEE INFOCOM*. 2010. IEEE.
39. Bartolini, N., et al., *Push & Pull: autonomous deployment of mobile sensors for a complete coverage*. Wireless Networks, 2010. **16**(3): p. 607-625.
40. Bartolini, N., et al., *Autonomous deployment of heterogeneous mobile sensors*. IEEE Transactions on Mobile Computing, 2011. **10**(6): p. 753-766.
41. Cheng, T.M. and A.V. Savkin, *A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks*. IEEE Communications Letters, 2009. **13**(11): p. 877-879.
42. Chang, C.-Y., C.-Y. Hsiao, and Y.-T. Chin. *The k-Barrier Coverage Mechanism in Wireless Mobile Sensor Networks*. in *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*. 2014. ACM.
43. Ma, H., et al., *Energy efficient k-barrier coverage in limited mobile wireless sensor networks*. Computer Communications, 2012. **35**(14): p. 1749-1758.
44. Kim, D., et al., *Maximum lifetime combined barrier-coverage of weak static sensors and strong mobile sensors*. IEEE Transactions on Mobile Computing, 2017. **16**(7): p. 1956-1966.
45. Wang, Z., et al. *Fault tolerant barrier coverage for wireless sensor networks*. in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. 2014. IEEE.
46. Wang, Z., et al., *Achieving k-barrier coverage in hybrid directional sensor networks*. IEEE Transactions on Mobile Computing, 2014. **13**(7): p. 1443-1455.
47. Zhang, L., J. Tang, and W. Zhang. *Strong barrier coverage with directional sensors*. in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. 2009. IEEE.
48. Han, R., L. Zhang, and W. Yang. *Maximizing strong barriers in lifetime-heterogeneous directional sensor network*. in *2016 International Symposium on Wireless Communication Systems (ISWCS)*. 2016. IEEE.
49. Khanjary, M., M. Sabaei, and M.R. Meybodi, *Barrier coverage in adjustable-orientation directional sensor networks: A learning automata approach*. Computers & Electrical Engineering, 2018. **72**: p. 859-876.
50. Mostafaei, H., et al., *A sleep scheduling approach based on learning automata for WSN partialcoverage*. Journal of Network and Computer Applications, 2017. **80**: p. 67-78.

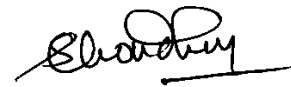
51. Mostafaei, H., M. Esnaashari, and M.R. Meybodi, *A coverage monitoring algorithm based on learning automata for wireless sensor networks*. arXiv preprint arXiv:1409.1515, 2014.
52. Mostafaei, H. and M.R. Meybodi, *Maximizing lifetime of target coverage in wireless sensor networks using learning automata*. *Wireless Personal Communications*, 2013. **71**(2): p. 1461-1477.
53. Esnaashari, M. and M.R. Meybodi, *Deployment of a mobile wireless sensor network with k-coverage constraint: a cellular learning automata approach*. *Wireless Networks*, 2013. **19**(5): p. 945-968.
54. Mostafaei, H., et al., *Barrier coverage of WSNs with the imperialist competitive algorithm*. *The Journal of Supercomputing*, 2017. **73**(11): p. 4957-4980.
55. Chen, J., J. Li, and T.H. Lai, *Energy-efficient intrusion detection with a barrier of probabilistic sensors: Global and local*. *IEEE Transactions on Wireless Communications*, 2013. **12**(9): p. 4742-4755.
56. Sheu, J.-P. and H.-F. Lin. *Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks*. in *2007 IEEE Wireless Communications and Networking Conference*. 2007. IEEE.
57. Kim, H., J.A. Cobb, and J. Ben-Othman, *Maximizing the lifetime of reinforced barriers in wireless sensor networks*. *Concurrency and Computation: Practice and Experience*, 2017. **29**(23): p. e4070.
58. Zhang, Y., X. Sun, and B. Wang, *Efficient algorithm for k-barrier coverage based on integer linear programming*. *China Communications*, 2016. **13**(7): p. 16-23.
59. Liao, W.-H., Y. Kao, and R.-T. Wu, *Ant colony optimization based sensor deployment protocol for wireless sensor networks*. *Expert Systems with Applications*, 2011. **38**(6): p. 6599-6605.
60. Huang, Y.-Y. and K.-Q. Li, *Coverage optimization of wireless sensor networks based on artificial fish swarm algorithm*. *Jisuanji Yingyong Yanjiu*, 2013. **30**(2): p. 554-556.
61. Maleki, I., et al., *A new approach for area coverage problem in Wireless Sensor Networks with hybrid particle swarm optimization and differential evolution algorithms*. *International Journal of Mobile Network Communications and Telematics (IJMNCT)*, 2013. **3**(6): p. 61-76.
62. Weng, C.-I., et al., *On-supporting energy balanced k -barrier coverage in wireless sensor networks*. *IEEE Access*, 2018. **6**: p. 13261-13274.

PLAGIARISM CERTIFICATE

1. We Dr Mukul Kumar Gupta (Internal Guide), Dr Sushabhan Choudhury (Co Guide/ External Guide) certify that the Thesis titled “Design and Implementation of an Energy Efficient k-barrier Coverage Network for Rural Applications” submitted by Scholar Mr/ Ms Vinay Chowdary having SAP ID 500043455 has been run through a Plagiarism Check Software and the Plagiarism Percentage is reported to be 9 %.
2. Plagiarism Report (First Page) generated by the Plagiarism Software is attached .



**Signature of the Internal Guide
Guide**



**Signature of External Guide/ Co
Guide**



Signature of the Scholar

ORIGINALITY REPORT

9%

SIMILARITY INDEX

3%

INTERNET SOURCES

7%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

www.ijmems.in

Internet Source

1%

2

University of Tennessee, Knoxville

Publication

<1%

3

Tri Gia Nguyen, Chakchai So-In. "Distributed Deployment Algorithm for Barrier Coverage in Mobile Sensor Networks", IEEE Access, 2018

Publication

<1%

4

Submitted to Visvesvaraya Technological University

Student Paper

<1%

5

www.mdpi.com

Internet Source

<1%

6

Lecture Notes in Computer Science, 2015.

Publication

<1%

7

onlineseouter.com

Internet Source

<1%

8

Simone Silvestri, Ken Goss. "MobiBar: An autonomous deployment algorithm for barrier

<1%
