Intelligent transportation system using wireless sensors networks

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electronics and Communication Engineering by Research

by

Aishwarya Jain
201231103
aishwarya.jain@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
June 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work in this thesis, titled INTELLIGENT TRANSPORTATION SYSTEM USING WIRELESS SENSOR NETWORKS by AISHWARYA JAIN, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser Dr. Garimella Ramamurthy

Dedicated to my parents.

Acknowledgments

The successful outcome of this thesis required a lot of support and encouragement from many people. I am fortunate to have guidance and assistance of all them, till the completion of my MS thesis work. I extend my gratitude to all of them.

First and foremost, I would like to thank my thesis advisor Dr. Rama Murthy Garimella. His guidance, knowledge, and support have helped me throughout my MS journey. I am grateful to him for allowing me to contribute to the current set of knowledge in this area. He kept me motivated throughout the whole process. He has played a key role in moulding my thesis and steering my thoughts in the correct direction.

I thank all my friends and well-wishers for their constant encouragement and helpfulness. I am fortunate to get assistance from my friends. Also, I would like to thank all my colleagues and lab members sincerely.

Lastly, I am thankful to my parents and brother, who supported and believed in me even in the most challenging phase. I express my deepest gratitude for providing unfailing support and continuous encouragement throughout my years of study.

Abstract

Wireless sensor networks are a network of small or large, dedicated sensors designed for particular and specialized purposes. These purposes can range from monitoring environmental behavior like temperature, pressure change, etc., to monitoring man-made physical objects and activities like vehicle speed, sound, density in a region, etc. Sensor networks have a vast application in fields that includes environmental monitoring, natural or artificial object tracking,

fire and water detection, natural calamity detection, and traffic monitoring. These groups of sensors and actuators collect information and data for the purpose they have been designed for and then send all the information to a central location for processing or general storage in databases for future use. Based on the topologies of the networks and routing protocols, a particular sensor network can be classified as hierarchical-based routing, location-based routing, and flat-based routing. These protocols are very simple and thus are very prone to attacks like Selective forwarding, HELLO flood attack, Sinkhole attack, Sybil attack, Wormholes, replaying routing information, and Acknowledgement spoofing or altering. Also, since one of the properties of these sensors is limited power, the wireless sensor networks consisting of such sensors generally face power shortages or outages altogether. Wireless Sensors that are small and limited battery capacity are widely used for several different applications.

For this reason, energy efficiency, as well as the cost of the overall network, are the two main factors influencing the Wireless Sensor Networks (WSN) performance. Low Energy Adaptive Clustering Hierarchy (LEACH) is an energy-efficient hierarchical-based routing protocol. The prime focus of this thesis is to analyze various existing LEACH protocols and to propose a new LEACH Protocol to resolve some of the issues mentioned above related to wireless sensor networks. This thesis presents an optimized cluster head selection mechanism protocol based on a weighted centroid approach. It is a clustering-based routing protocol that minimizes total energy consumption. Our LEACH-WC protocol algorithm outperforms other LEACH protocol algorithms by making nodes offer themselves candidates for high-energy cluster-heads and adapting the corresponding clusters based on the nodes that choose to be cluster-heads at a given time. We have also discussed an exciting, combined cost and energy wastage minimization problem, and an interesting solution is proposed.

Along with the improvement in the existing LEACH protocol, we have also discussed an approach that indexes the physical location of various entities across the globe. In the future, combining this improvement in LEACH protocol with the proposed indexing system approach and their application may result in an even better and more intelligent transportation system using wireless sensors network.

Contents

Chapter 1

Introduction

1.1 Wireless sensor networks

Sensor Nodes, also known as a mote, are part of heterogeneous systems called sensor networks. They are small, lightweight, and portable, equipped with a transducer, microcomputer, transceiver, and power source.

The transducer generates electrical signals based on the physical situation that it senses. The microcomputer stores the sensed information after it processes it. The transceiver sends data to it based on instructions from the base station or a central computing system.

The constraints in the size and cost result in constraints on energy, memory, processing speed, and bandwidth for communication.

Wireless Sensor Networks are characterized by the following:

- They have constraints over how much power they can have at a particular moment.
- They have some limited capability to deal with the failing nodes.
- The nodes are not homogeneous.
- They are generally deployed in large areas and large numbers
- The nodes are not static.
- They sometimes can face issues related to communication.
- They have a very dynamic network topology.
- They can sustain grinding natural situations like floods, earthquakes, etc.

1.2 Applications of Wireless sensor networks

The advantages of WSNs are that they can incorporate various reasons to accomplish various tasks, starting from magnetic, thermal, visual, seismic, infrared, and radar Sensors.

These sensor nodes can be used for continuous sensing, location sensing, motion sensing, and event detection.

Some examples of the applications of WSNs are:

1.2.1 AREA MONITORING APPLICATIONS

In Area Monitoring Applications, the WSNs are deployed over a region where some physical activity is to be monitored via sound/vibrations. If an event is detected, it is reported to the base station, which takes appropriate action.

Such networks can extend their usage to traffic control systems to detect the presence of a high-speed vehicle. Another application of the WSN could be in the military, where we need to observe an area.

1.2.2 Environmental Applications

Some examples include forest fire detection, flood detection, air pollution checks, and tracking animals and birds.

1.2.3 Health Applications

WSNs can be used in hospitals' integrated patient monitoring, diagnosis, and drug administration.

1.2.4 Industrial Applications

WSNs are now widely used in industries like Machinery-based maintenance. They could be used in previously inaccessible locations or hazardous locations. They can detect water levels within all ground wells and watch out for water percolated through a solid, leaching out some of the constituents and their subsequent removal.

1.2.5 Other Applications

WSNs can also be used in our day-to-day life applications. They can be used in Vacuum cleaners, Microwave ovens, VCRs, and refrigerators.

1.3 Routing Protocols

Depending on the network's topology, routing in wireless sensor networks can be classified as location-based, hierarchical, and flat-based.

- In flat-based routing, all the nodes in the topology are assigned the same functionality or role.
- In hierarchical-based routing, different functionalities and roles are assigned to the nodes based on their hierarchy.
- In location-based routing, the path is usually determined by considering the sensor node's position in the field.

Depending on the protocol operation, routing protocols can be classified into multipath-based, negotiation-based, QoS-based, coherent-based routing, or query-based.

- In coherent-based routing, the data is aggregated with minimum processing before forwarding. Here, energy efficiency is achieved by path optimality.
- In negotiation-based routing, the sensor nodes negotiate with the other nodes to eliminate unnecessary communication by using negotiation. Communication decisions are made based on available resources.

- In multipath-based routing, multiple alternative paths are used to improve the network performance, i.e., fault tolerance, increased bandwidth, improved security, energy efficiency, and reliability.
- QoS-based routing maintains a balance between the energy consumed and the data quality.
- In query-based routing, a station sends a query to find the specific events among wireless sensor networks. Usually, these queries are executed in either natural language or high-level query language.

Routing protocols can be classified into proactive, reactive, and hybrid, depending on how a source can find a route to the destination.

- In proactive protocols, routes are determined even before they are required.
- In reactive protocols, routes are determined only when they are required.
- Hybrid protocols are the amalgamation of proactive protocols and reactive protocols.

1.4 Problem statement

Our primary target is to propose a new LEACH Protocol which is better than existing LEACH protocols. We also tried formulating an exciting problem on combined cost and energy wastage minimization and proposing an interesting solution. We have discussed distributed computation and communication of minimum and maximum sensed value over a uniform rectangular grid. Event detection over a distributed wireless sensor network is also discussed. We also tried proposing an approach for indexing various physical locations on the planet.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 briefly introduces the LEACH Protocol and its various variations.

Chapter 3 proposes a new LEACH protocol based on a weighted centroid algorithm.

Chapter 4 discusses an interesting, combined cost and energy wastage minimization problem and its exciting solution.

Chapter 5 discussed an approach that indexes the physical location of various entities across the globe.

Chapter 6 is the concluding chapter, in which we discuss our contributions to the community, the limitations of our work, and the scope of future work.

Chapter 2

LEACH Protocol

2.1 Introduction

Heinzelman [1] introduced Low Energy Adaptive Clustering Hierarchy (LEACH) as a hierarchical routing algorithm for wireless sensors network to expand the network's lifetime. LEACH distributes all the sensors into groups of sensors or nodes, which are individually called clusters. A cluster head is then chosen for each of such clusters. All the nodes in a cluster collect the required information, according to their specific design and purpose, first and then send all the collected information and data to the cluster head. Cluster-head, in turn, collects all the information from all the nodes in its clusters, aggregates them, and sends them to the base station. Since the energy required to send information to the base station is more than the energy required just to send information to the cluster head, the cluster head generally loses power faster. Thus, there arises the requirement to rotate cluster-head among all the nodes in a cluster. This rotation can be done randomly, in a well-defined pattern, and based on well-defined algorithms. Each such algorithm then results in various degrees of improvements in the energy efficiency and prolongment in the life of such wireless sensors network.

2.2 Operation,

The working of a LEACH operation, can be understood by dividing them into two cycles:

1. Setup cycle
2. Steady cycle

In the setup cycle, all the sensors are divided into groups of clusters, and then from each cluster, a node is chosen as cluster-head. On the other hand, in a steady cycle, in a particular cluster, data is collected by the cluster head from all the nodes and is then sent to the base station. Since the steady cycle comprises the main functionality and purpose, it is kept longer than the setup cycle.

1. Setup cycle: In the setup cycle, once the cluster head is decided in a particular cycle, it sends a message or a signal to all the nodes in its network, asking them to join their cluster. The non-cluster head node decides whether it wants to join the cluster based on the strength of the signal sent by the cluster head. If they find a signal to be strong enough, they communicate back with the cluster head that they will be part of their respective cluster. They do this by sending them an acknowledgment message. Cluster-head, in turn, on the number of nodes willing to be part of its cluster and the type of information it needs to send to the base station, creates a TDMA schedule. This TDMA schedule dictates the time slot for each node, in which they need to send the required and sensed information to the cluster head. If the cluster size becomes too big, the cluster head can choose another cluster head. This old cluster-head will get a chance to become cluster-head again once all the nodes in the cluster get a chance to become cluster-head, at least once.

2. Steady cycle: In the steady cycle, non-cluster heads start sensing data and sending it to the cluster head. Cluster-head, in turn, collects the data from all the nodes in its cluster and sends it collectively to the base station.

After a predefined time, the sensor network returns to the setup cycle to choose a new cluster head, continuing until the whole network is drained of completed energy.

2.3 Assumptions

Some assumptions are considered while analyzing the LEACH protocol. These assumptions may or may not affect the actual wireless sensor networks practically. Some of these assumptions are as follows:

1. All nodes have enough power to send data to the base station directly. This factor will weigh in when cluster heads are chosen as the nodes that need more power should not be considered while selecting cluster-head.
2. Each node should have enough computation power to support different MAC protocols.
3. Each node senses some data it needs to send to the cluster head, i.e., nodes always have some data it needs to send.

2.4 Variations

Over time, several variations of the LEACH protocol have appeared. These variations have improved over the original LEACH protocol. These improvements can vary from the long life of the wireless sensor networks to the protocols which add more security measures to the overall network. Some of such variations are as follows:

2.4.1 LEACH-Centralized

LEACH-Centralized [2] is an updated version of LEACH. In this method, the base station asks all the nodes for their remaining energy details. After all, nodes send this detail to the base station; the base station will show which node will be the cluster head. The base station chooses cluster heads based on the remaining energy of the nodes. The main disadvantage of the Leach-C protocol is that its performance decreases when the energy used for communicating with the base station (for giving the energy detail and then choosing the next cluster head) is very high or more.

2.4.2 LEACH-E

Leach-E [3] is an extended protocol for selecting cluster heads based on the remaining energy and distances between sensor nodes. This increases the lifetime of the given WSN network. In the general leach protocol, the random distribution of nodes will generally result in some nodes having higher or lower energy than others during the cluster head rotation process. This unevenness can shorten the life cycle of the WSN network as we will still not be entirely using every node’s potential to its full extent. There is a kind of uneven load distribution among the nodes in one cluster to the fair use of the entire node’s energy. However, it does not satisfy the potential use of node energy and good performance metrics.

2.4.3 I-LEACH

Both the residual energy and distance between the nodes are considered in ILEACH [4]. It is an improved algorithm that improves many aspects, like improved packet delivery ratio, better average delay, more efficient throughput, and more nodes alive. The other cluster nodes' remaining energy is considered at each round. The selection of cluster head selection is based on the signal strength received from the sink and energy left in the node. The drawback is the creation of more clusters and forming of big and small clusters.

2.4.4 Node Ranked–LEACH

In this protocol [5] selection of cluster head is based on the link connection number with the remaining energy of the other nodes and their respective distance from BS. By decreasing the packet loss, results are improved. For the distribution of cluster heads, this algorithm is very effective. Significantly less energy is needed for communicating between various nodes and base stations. Compared to other LEACH versions, it is done by reducing the energy needed to ask for redundant data regarding the residual energy details.

2.4.5 Levelling and Sectoring LEACH

It describes various data-centric routing protocols and proposed protocols known as leveling and sectoring [6] to reduce energy consumption. In a homogeneous network where every node acts as a cluster head and sends the data directly to the sink, Leveling and Sectoring LEACH algorithm[6] proposes an approach in which instead of sending data to the sink, it will send the data from one level to another and stop reverse propagation of packet and save an exponential amount of energy.

2.4.6 Greedy LEACH

In this protocol [7], all the sensors are randomly deployed by drone or plane. All the sensors know their location and neighboring nodes' location by sending the message that all are homogenous nodes. The sensor is divided into sectors. Each sector will select the cluster head based upon the following constraint: weight and energy consumption calculation through different routing paths to ensure the selected cluster head is optimal. The cluster head will communicate with each other by connecting themselves in the form of chains by using greedy algorithms. The transmission power is adjusted based on the distances between the sensor nodes, and the mobile sink is allocated to every sector.

Chapter 3

Wireless sensor network using Weighted Centroid-based Leach Protocol

3.1 Introduction

In this chapter, we quantitatively show the energy savings obtained in a heterogeneous wireless sensor network using the Weighted Centroid algorithm. Due to the energy constraints on the sensor nodes in hierarchical networks, efficient cluster head selection has become crucial. We propose an algorithm in which all the nodes do not necessarily have the same initial energy. Our main contribution is to derive the location of a weighted Centroid, which minimizes the weights sum of squares distances from the sensor nodes to the cluster head location. Analytical results are validated through simulation in OMNET++ using various performance metrics and compared with different cluster head selection protocols such as I-Leach, Leach-E, and NR-Leach. This shows the promising potential of our Weighted Centroid algorithm.

The technology of wireless sensor networks was innovated to monitor and control various natural/artificial phenomena. The sensor nodes are deployed in the following ways [11]:

- Planned deployment
- Random deployment.

For instance, in an application such as forest fire monitoring, the sensor nodes are randomly deployed (from, say, a helicopter). Conserving the energy (in the battery) consumption and prolonging the network lifetime is an important factor in designing protocols at various network layers (OSI model).

LEACH is an energy-efficient protocol based on clustering the sensor nodes and utilizing only the cluster heads for communicating sensed information to the base station, as discussed earlier. In most of these versions of LEACH, it was implicitly assumed that the cluster members are located one hop (in terms of wireless communication radius) from each other. However, this assumption gets violated in many real-world deployments of WSNs. In such real-world WSNs, the distance profile of sensor nodes must be considered while selecting the cluster head.

The research work reported in this chapter is based on the optimization interpretation of the centroid of a finite number of pattern vectors. It is specifically shown that the location of the centroid (as the vector whose components are the mean of respective components of pattern vectors) minimizes the sum of squared values of distances (from the pattern vectors to it). This result was used to locate the cluster head (closest to the centroid) when the residual energy in all sensors is the same. However, we realized that when the residual energy of all sensors is not equal, we need to weigh the distances with weights related to residual energies. The associated optimization problem is formulated and solved in this chapter.

3.2 General Derivation

In this section, we consider the case where a real number weights the distance from a pattern vector to the centroid. The sum of squared weighted distances from the pattern vectors to the centroid is minimized.

In the following section, we used the weighted centroid as the position of the cluster head in every cluster.

Derivation: For N nodes in M-dimensional plane

Patterns:$\{(x11,x12,x13,\ldots.x1m),(x21,x22,x23\ldots.x2m)\ldots\ldots..(xN1,xN2,xN3,\ldots xNM)\}$

Centroid: $(\alpha 1,\alpha 2,\alpha 3,\ldots.\alpha M)$

Let’s assume respective weights to be $(a1,a2,a3,\ldots.aN)$

$J(\alpha 1,\alpha 2,\alpha 3,\ldots.\alpha M) = a1\{(x11 - \alpha 1)2 + (x12 - \alpha 2)2 + \ldots..+ (x1M - \alpha M)2\} + a2\{(x21 - \alpha 1)2 + (x22 - \alpha 2)2 + \ldots..+ (x2M - \alpha M)2\} + aN\{(xN1 - \alpha 1)2 + (xN2 - \alpha 2)2 + \ldots..+ (xNM - \alpha M)2\}$

$$\frac{\delta J}{\delta \alpha 1} = -2a1\,(x11 - \alpha 1) - 2a2\,(x21 - \alpha 2) - \ldots\ldots..-2aN\,(xN1 - \alpha 1)$$

$$\frac{\delta J}{\delta \alpha 2} = -2a1\,(x12 - \alpha 2) - 2a2\,(x21 - \alpha 2) - \ldots\ldots..-2aN\,(xN2 - \alpha 2)$$

Similarly,

$$\frac{\delta J}{\delta \alpha M} = -2a1\,(x1M - \alpha M) - 2a2\,(x2M - \alpha M) - \ldots\ldots..-2aN\,(xNM - \alpha M)$$

$$\sqrt{H(\alpha_1\ ,\alpha_2\ ,\ldots.,\alpha_M\ )}\ =\ \sqrt{\begin{bmatrix} \frac{\partial^2\ J}{\partial {\alpha_1}^2} & \cdots & \frac{\partial^2\ J}{\partial\alpha_1\ \partial\alpha_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\ J}{\partial\alpha_M\ \partial\alpha_1} & \cdots & \frac{\partial^2\ J}{\partial {\alpha_M}^2} \end{bmatrix}}$$

$$\sqrt{H(\alpha_1\ ,\alpha_2\ ,\ldots.,\alpha_M\ )}\ =\ \sqrt{\begin{bmatrix} 2a_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2a_N \end{bmatrix}}$$

$$\frac{\delta J}{\delta\alpha_1} = 0 \qquad \frac{\delta^2 J}{\delta\alpha_1\,\delta\alpha_2} = 0 \quad \text{for } i \neq j$$

$$-2a_1\,(\,x_{11} - \alpha_1)\ -2a_2\,(\,x_{21} - \alpha_1)\ -\ \ldots\ldots{-}{-}2a_N\,(\,x_{N1} - \alpha_1) = 0$$

$$\alpha 1 = \frac{a_1\ x_{11} + a_2\ x_{21} + \ldots.. + a_N\ x_{N1}}{N}$$

$$\alpha_2 = \frac{a_1\ x_{12} + a_2\ x_{22} + \ldots.. + a_N\ x_{N2}}{N}$$

Similarly

$$\alpha_M = \frac{a_1\ x_{1M} + a_2\ x_{2M} + \ldots.. + a_N\ x_{NM}}{N}$$

(α1,α2,α3,….αM) : Decided based on weighted mean

a1 = a2 = ……= aN =1

Will give centroid as (α1,α2,α3,….αM)

3.3 Algorithm and Flowchart

We propose an algorithm, in which all the nodes do not necessarily have the same initial energy. Our main contribution is to derive the location of a weighted Centroid, which minimizes the weights sum of squares distances from the sensor nodes to the cluster head location.

Step-1: Consider sensor nodes with different energy levels.

Step-2: The distance from a pattern vector to the centroid is weighted by a real number.
Step-3: Assume the number of nodes is N in the M dimensional plane.
Step-4: Centroids of the clusters are calculated by using CC.
Step-5: If a sensor node has larger residual energy, it can afford to be at a larger distance from the cluster head. On the other hand, if a sensor node has smaller residual energy, it must be chosen to be closer to the cluster head.
Step-6: Select the node nearer to the smaller residual energy nodes and nearby Centroid as Cluster Head.
Step-7: If the cluster head energy is reduced below the threshold, it will not be considered in the cluster head selection process.

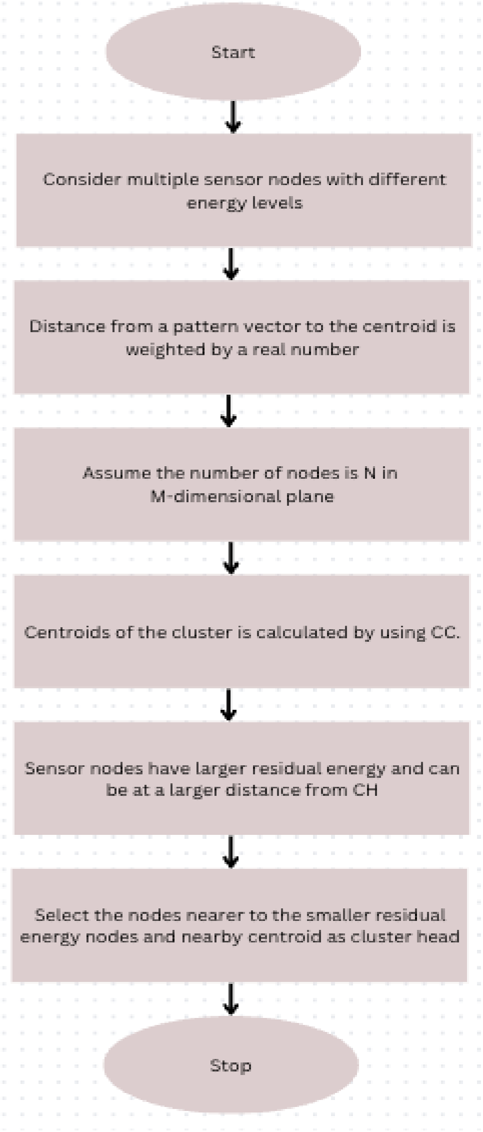


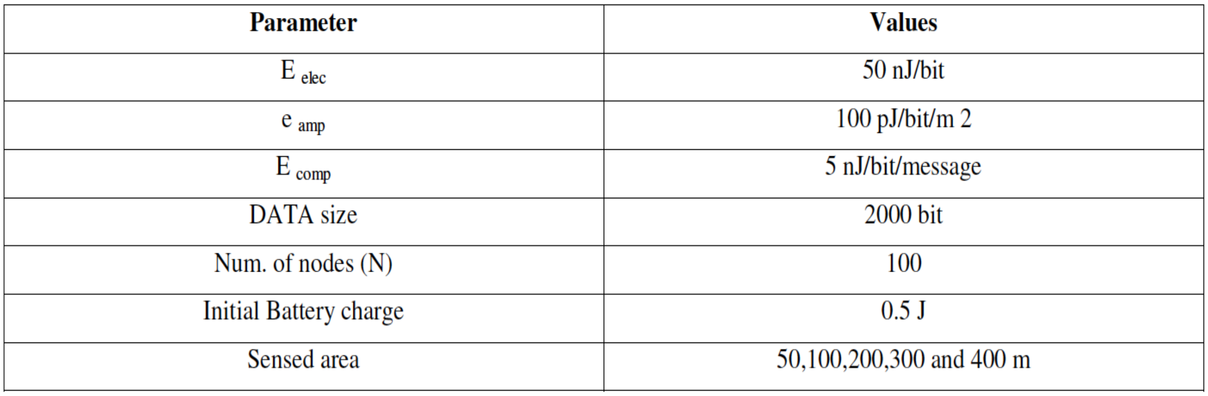
Fig 1. Flowchart Constrained Clustering Weighted Centroid

3.4 Simulation and Results

The proposed weighted centroid algorithm is evaluated using the OMNET++ simulator in which the cluster head selection is made based on their weights. The following assumptions have been considered: In each round, only one transmission is performed to CH or Sink. Assume the network is heterogeneous, where all nodes have different energy. The sink location is unknown to all the sensor nodes, assuming no collision in cluster formation. The comparison is made between existing approaches such as I-LEACH, LEACH-E, and NR-LEACH and the proposed LEACH-WC protocol. The simulation is carried out by varying the coverage range of the sensor nodes, as shown in Table 2, by considering various performance metrics such as the first node die, the last node die, and the number of rounds.

For these simulations, the following parameters have been used:

Table 1. Simulation parameters



Parameter	Values
E_{elec}	50 nJ/bit
e_{amp}	100 pJ/bit/m 2
E_{comp}	5 nJ/bit/message
DATA size	2000 bit
Num. of nodes (N)	100
Initial Battery charge	0.5 J
Sensed area	50,100,200,300 and 400 m

The results obtained by simulating various protocols by varying the edges (i.e., coverage region of sensor nodes) and recording the scalar and vector files are as shown below.

Table 2. Simulation Result of Various Leach Protocols

Edges	Metrics	I-Leach	Leach-E	Leach-NR	Leach-WC
50	First node	970	2149	2136	2313
	End node	8004	6468	7347	7290
	Rounds	2668	2156	2449	2430
100	First node	376	833	892	980
	End node	3246	2529	3030	3000
	Rounds	1082	843	1010	1000
200	First node	103	227	291	270
	End node	969	741	1041	933
	Rounds	323	247	347	311
300	First node	41	105	97	123
	End node	447	351	396	471
	Rounds	149	117	132	157
400	First node	13	56	54	62
	End node	273	210	237	252
	Rounds	91	70	79	84

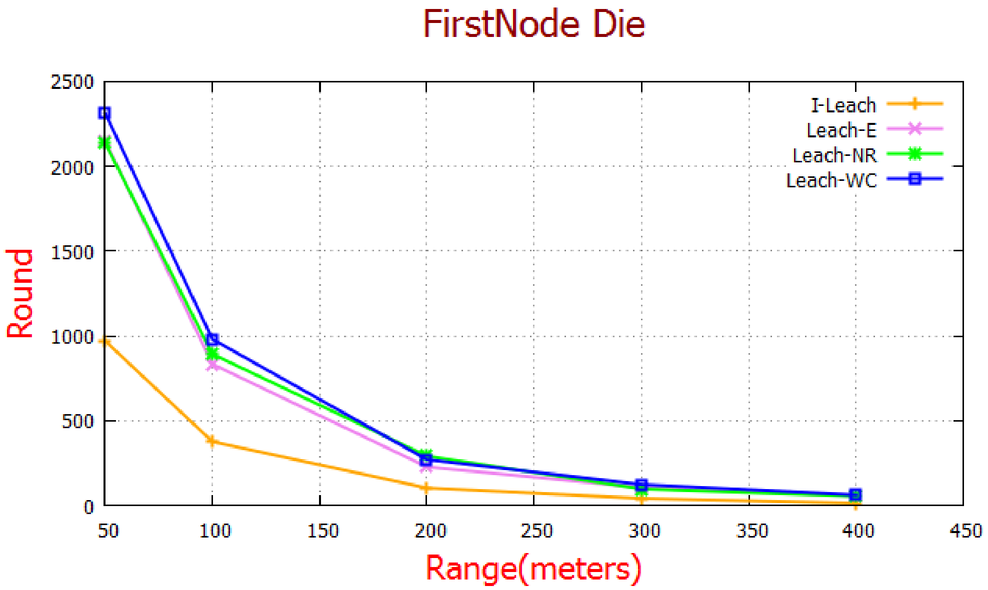


Fig 2. First Node Die

From Fig: 2 and Table 2 reading values, it can be seen that the first node die time is more in the proposed weighted centroid algorithm when compared with various latest versions of the Leach protocols. The simulation has been tested by varying the coverage region of the sensor nodes, and it has been observed that the first node die time is always more in the proposed algorithm.

Fig 3. Last Node Die

From Fig: 3 and Table 2 reading values, it can be seen that the end node die time is almost more in the proposed weighted centroid algorithm when compared with various latest versions of the Leach protocols. The simulation has been tested by varying the coverage region of the sensor nodes, and it has been observed that end node die time is almost more in the proposed algorithm.

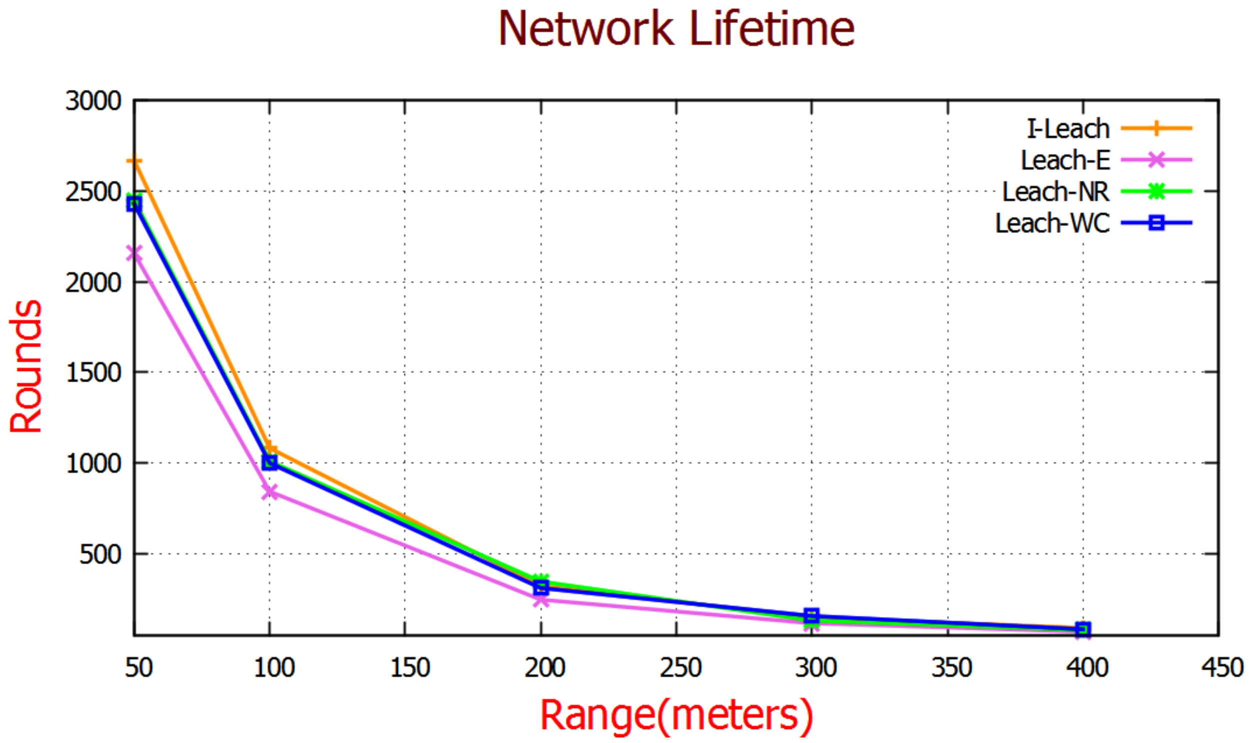


Fig 4. Network Lifetime

In Fig: 4 and Table 2 reading values, it can be seen that the Network lifetime is more in the proposed weighted centroid algorithm when compared with various latest versions of the Leach protocols if the coverage region of the nodes is high. If the coverage region is reduced to a smaller value, other protocols have almost the same network lifetime as the proposed algorithm.

3.4 Conclusions

In this chapter, we present an optimized cluster head selection mechanism protocol termed weighted centroid, a clustering-based routing protocol that minimizes total energy consumption. Our LEACH-WC protocol algorithm outperforms other LEACH protocol algorithms by making nodes offer themselves candidates for high-energy cluster-heads and adapting the corresponding clusters based on the nodes that choose to be cluster-heads at a given time. Distribution of the energy between the nodes in the network is an efficient way to reduce energy wastage from a global perspective and increase system lifetime. We formulated and solved the problem of finding the location of a weighted centroid (when some weights weigh the distances from it to the sensor nodes), which minimizes the sum of squared weighted distances from it to the sensor nodes. Simulations show that our Weighted Centroid algorithm reduces communication energy compared to other LEACH protocol algorithms. For testing our assumptions about our LEACH-WC algorithm, we are using the network simulator [12] OMNET++ which is well known to perform better simulation in the field of WSN when compared to other network simulators. We simulate our LEACH protocol algorithm and other LEACH algorithms in OMNET++. This comparison will test the efficiency of our algorithm and give us a more accurate state of the advantages and disadvantages of the different protocols compared to ours. Based on our OMNET++ simulations described above, we are confident that the proposed LEACH-WC protocol will outperform conventional communication protocols.

3.4 Future work

To implement the proposed LEACH-WC protocol for other paradigms, such as MWSN, can be done by considering different types of deployment processes, mobility of the nodes, and comparison with the existing mobility-based routing protocols.

Chapter 4

Wireless sensor networks on uniform grid

4.1 INTRODUCTION

Wireless Sensors that are small in size and have limited battery capacity are widely used for several applications. For this reason, energy efficiency, as well as the cost of the overall network, are the two main factors influencing the Wireless Sensor Networks (WSN) performance. This chapter proposes energy-efficient placement of sensors in a Wireless Sensor Network on the uniform rectangular grid (as a sensor field). Also, distributed computation of minimum and maximum over such sensor fields is discussed (for minimizing computations and communications delay).

Advances in VLSI and MEMS technologies have enabled the manufacture of tiny wireless sensors that transmit the sensed information wirelessly to a sink node called a Base Station. The technology of wireless sensor nodes is projected as one of the ten technologies that will change the world. Cyber-physical systems will enable the Internet of Things (IoT) to materialize with the aid of Wireless Sensor Networks (WSNs). Research efforts on WSNs are already beginning to be materialized in real-world networks [13]. One feature that distinguishes WSNs from other computer or communication networks is network-distributed computation. In such a distributed system, the sensor nodes perform computation locally on the sensed data and the data received from other sensor nodes. Hence, the problem of efficient distributed computation naturally arises in designing and implementing WSNs [14].

In many exciting applications of WSNs, the network's topology is under the user's control. The sensor field in many applications constitutes a uniform/regular rectangular grid. The sensors are located at the grid points. In such a network architecture, one of the research problems attempted Cost-Optimal placement of 2-types of sensor nodes (with different costs) (i.e., the goal is to minimize the total cost of placing the sensors on the grid). Such a sensor placement problem naturally leads to an integer linear programming problem [10][15][16]. Researchers realized that in many applications, it is not possible to replace the sensor when the battery is dead manually is impossible. Hence, minimization of energy wastage is an important design consideration. It was realized that due to redundancy in sensed readings of sensors located closely, it is only necessary for one of the nodes to collect locally sensed data and transmit the processed readings to the Base Station. Thus, Low Energy Adaptive Clustering Hierarchy (LEACH) and several variants are proposed for energy-efficient routing of sensed data [15][17]. Hence, we realized that along with cost minimization, energy efficiency is an important design consideration in WSNs. Thus, in this chapter, we propose efficient algorithms for minimizing energy wastage in transmissions/receptions of sensed data by sensors [18][19][20].

This chapter is organized as follows. Section 5.2 discusses distributed computation or communication of minimum or maximum over a uniform grid. Section 5.3's results of Section 5.2 are generalized to primitive recursive functions. Section 5.4 discusses sensor placement over a uniform grid by minimizing energy and cost. In Section 5.5, event detection over distributed wireless sensor networks is discussed and later followed by a conclusion in Section 6.6.

4.2 Distributed Computation/Communication of minimum/maximum over uniform grid

We now consider a regular rectangular grid in which a sensor is located at each grid point, i.e., we consider such a Wireless Sensor Network (WSN).

4.2.1 Goal

Efficiently compute the minimum and maximum value of all sensed readings. Also, communicate the minimum and maximum value of all sensed readings to the Base Station at the top of the grid.

4.2.2 Efficient Computation and Communication

We consider computation and Communication to occur concurrently along each line (in the grid). Also, each node maintains a 2-tuple containing the current maximum and minimum in the format: (Current Maximum, Current Minimum): On receiving a reading from a neighboring node down the line, i.e., every node computes the current maximum as well as minimum and communicates the tuple to the node up the line. It readily follows that the number of comparisons is 2N - 3. This number comes out since the tuple at the last node contains the same number in both the positions of 2-tuples. Also, the node up the line to the last node only makes one comparison, and all other N - 2 nodes perform two comparisons (2(N - 2) + 1 = 2N - 3). Further, the delay in communicating the 2-tuple (i.e.(Max Value, Min Value)) to the base station Nd, where d is the delay on each wireless link-Without the 2-tuple idea, the number of comparisons will be 2(N - 1) = 2N - 2: Also, if maximum and minimum are not concurrently computed the delay will be 2N’(d) = 2Nd.

4.3 GENERALIZATION TO PRIMITIVE RECURSIVE FUNCTIONS

It is well known that the Minimum and Maximum functions, i.e., f(x1, x2,….., xN) (where (x1, x2,...., xN) are variables. For example they can represent the temperature values on the sensor field) are Primitive Recursive Function in the sense that f(x1, x2,....,xN+1) = f(f(x1, x2,...., xN) xN+1) explicitly Min(x1, x2,...., xN), Max(x1, x2,...., xN) are necessarily primitive recursive functions.

Claim: On a uniform, rectangular grid, the algorithm for computing maximum and minimum in a parallel and distributed manner naturally generalizes to any primitive recursive function. We now realize that by determining the maximum and minimum of, say, temperature measurements on the sensor field, we can readily determine their difference with one more arithmetic operation, Thus, the dynamic range of temperature can be readily determined. Now, we focus our attention on computing the average temperature of measurements on the sensor field, i.e.,

$$g(x_1, x_2, ...x_N) = \frac{(x_1 + x_2 + ... + x_N)}{N}.$$

It readily follows that g(x1, x2, ..., xN) is not a primitive recursive function since g(x1, x2, …., xN+1) is not equal to g(xN+1, g(x1, x2, ..., xN): But, the related function h(x1, x2, ..., xN) = x1 + x2 + ... + xN is indeed a primitive recursive function. Hence, we employ the following 2-tuple idea to determine the average value recursively. The 2-tuple that gets propagated towards the base station is given by (h(x1, x2, ..., xN), N). With the tuple being propagated towards the base station, we determine the average temperature using the following simple computation (performed finally).

$$g(x_1, x_2, ..., x_N) = \frac{h(x_1, x_2, ..., x_N)}{N}.$$

Note: In the above sense, the (Average/function) can be defined to be a modified recursive function.

4.3.1 Concurrent Computation of Primitive Recursive Functions

In a practical application, a uniform rectangular grid-based sensor field is distributed over a large geographical region. As discussed earlier, minimum/maximum computation and Min - Max and Max-Min values naturally arise in predicting events such as fire and snowfall over the sensor field. Prediction of such extreme events needs to be performed in real-time, and the delay in computing the maximum and minimum should be low. Toward the reduction delay involved, we propose the following concurrent computation idea over a vast uniform rectangular grid.

For instance, consider a ‘100 x 100’ grid. Divide the grid into ‘4, 50x50’ grids and locally compute the maximum/minimum temperature (using the tuple idea proposed in Section 2). These local minimum/maximum values on four grids are exchanged in real-time, and the global maximum/minimum value over the ‘100 x 100’ grid is determined. It can easily be reasoned that the number of comparisons will not get reduced (at most, one less comparison). Still, concurrent computation over the entire sensor field significantly reduces the communication delay (using the tuple idea) [23]. The number of sub-blocks into which the sensor field is divided is based on the size of the grid and the wireless/wired connectivity between the local Base Stations.

In the above discussion, the 2-tuple idea, i.e. (Minimum, Maximum), is proposed to simultaneously determine the maximum and minimum temperature over the sensor field (by propagating the 2-tuple from the node farthest to the Base Station towards the node closest to the Base Station on all columns of the rectangular grid). Finally, the 2-tuples are merged at the Base Station to determine the maximum and minimum over the entire sensor field. As a natural generalization, we can compute the 4-tuple (Minimum, Second Minimum, Second Maximum, and Maximum) to account for outliers corrupting the maximum, and minimum temperature values, i.e., Second maximum and Second Minimum may not be corrupted by measurement noise. The above 4-tuples are propagated parallelly along the columns (of the rectangular grid) towards the Base Station. The resulting 4-tuples received at the Base Station must be merged to determine the maximum/minimum over the entire sensor field.

Note: As discussed earlier, the average temperature over the entire sensor field can also be determined recursively (again using, say 5-tuple idea) as in the case of maximum/minimum. The weighted average can also be computed recursively.

4.3.2 Data-Centric Queries: Efficient Computation of Desired Order Statistics

It is reasonably assumed that base stations (distributed over a large geographical region) are constrained by t power/energy (battery) in wireless sensor networks. Hence instead of using the K-tuple idea (K 2), the base station broadcasts the minimum/maximum temperature value over the Base Station. Hence, all the nodes, particular nodes with relevant minimum/maximum values, know the critical desired values. Such node/nodes do not participate in comparison operations in the next round. Thus the second minimum/maximum can be computed with N-1 or fewer comparisons (with one or more nodes not participating in comparisons).

4.4 SENSOR PLACEMENT ON UNIFORM GRID COST

4.4.1 Optimal Placement of sensors through ILP: Algorithm-1

It is shown that the problem of cost-optimal placement of costly and cheap sensors (2 types) over a uniform grid can be formulated as an Integer Linear Programming problem (ILP). We now discuss a greedy algorithm to minimize computational complexity.

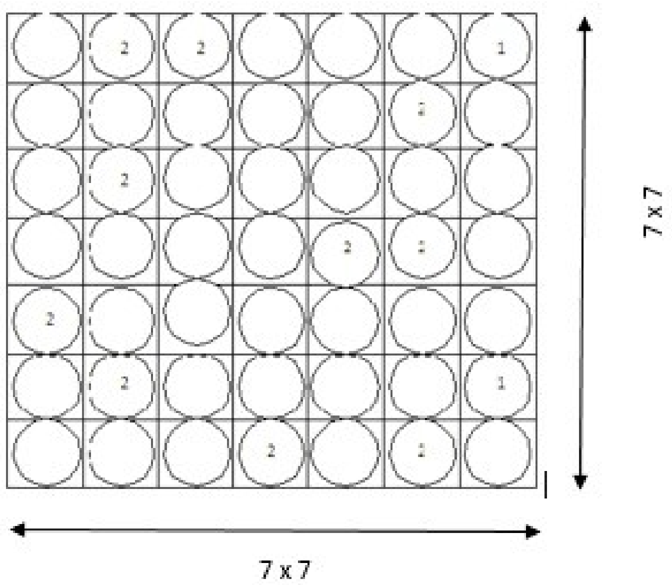


Fig. 1. 7*7 Grid with Type1 and Type2 sensors

4.4.2 Greedy Approach: Algorithm-2

With the help of a greedy approach, where the ‘N x N’ grid can be formed using the 4 ‘N/2 x

N/2’ matrix. As shown in Figure, ‘4, 14 x 14’ matrices are used to make one ‘28 x 28’ matrix. Therefore from a smaller grid solution, we could form a solution of a large grid. Although the solution will not be optimal, it will provide a correct solution, i.e., all points will be covered by at least M minimum number of sensors, and the algorithm can be implemented on any grid dimension. From the above discussion, we can determine the placement of costly and cheap sensors to minimize the total cost under some conditions.

We now have an algorithm to minimize energy wastage in unnecessary transmissions/receptions from sensor nodes. In the spirit of the leveling and sectoring algorithm, it is most logical to choose the sophisticated (and costly) sensors as the cluster heads, i.e., from the solution to cost-optimal sensor placement problem (solved using 0-1 linear programming), we arrive at the location of cluster heads. We now divide the sensor field (uniform rectangular grid) into clusters, whereby members of a cluster transmit the sensed information to the local cluster head.

Based on the location of the Sink/Base Station, the way levels are decided [21]. Fig 2 gives the divide and conquer approach for getting optimal solutions. Fig.3, Fig. 4, and Fig.5 indicate the various Leveling schemes used based on the Sink/Base Station placement. That way, levels are decided, as illustrated in the following figures. The concentric levels are decided so that based on the coordinates of the grid points, their level ID can easily be determined. Thus every sensor (including cluster heads, i.e., costly sensors) readily knows the level and position of the Base Station. Now, we minimize energy wastage in the following manner. i) Cheap sensors in a cluster transmit the sensed information only to the local cluster head. ii) Based on the level numbering, every cluster head at any level transmits data to the BS from a cluster head at a higher level number only. Packets received from the cluster heads at lower level numbers are dropped since those transmissions are progressing in the wrong direction. This directed broadcast algorithm readily minimizes energy wastage in transmissions and receptions. The actual quantification of energy savings in transmissions and receptions can easily be derived and is avoided for brevity [22].

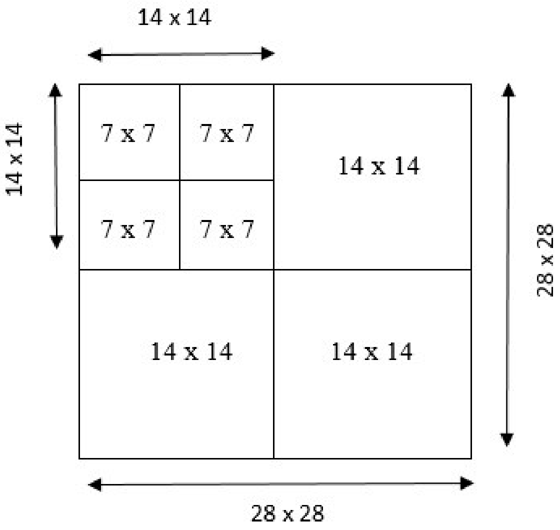


Fig. 2. 28 x 28 grids can be formed using 4 14 x 14

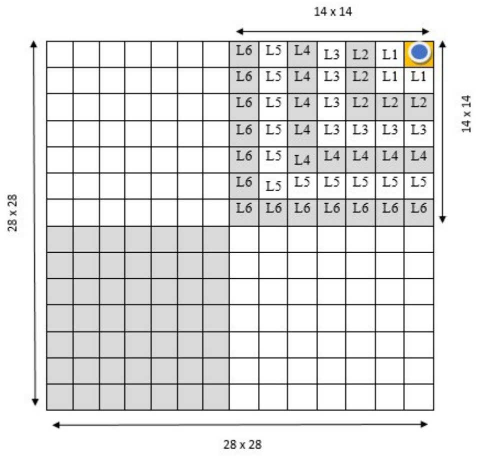


Fig. 3. WSN Grid Size 28*28(4 N/2*N/2) divided into levels

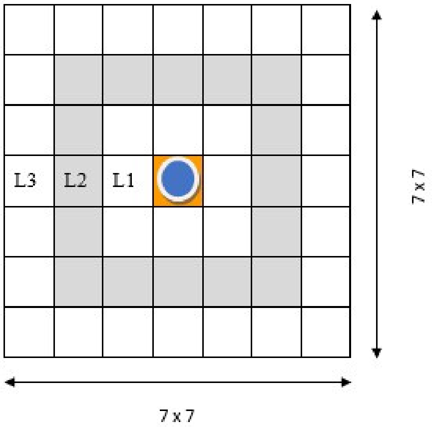


Fig. 4. Representation of leveling in 7*7 WSN Uniform Grid

4.5 Distributed wireless sensors networks: Event detection and simulation results

We now consider several uniform rectangular grids (sensor field) based WSNs deployed over a large geographical region. Distributed Mean and Median computation is performed simultaneously on all the connected WSNs. The Base Stations can communicate with each other efficiently. Fig. 5. Leveling with the number of sensors in 14*14 WSN Matrix For instance, consider a Weather monitoring (particularly temperature monitoring) WSN deployed over a city in a local country. We want to predict snowfall. For such event detection problems, the base stations (each of which computes minimum temperature locally) compute the maximum of all local minimum temperature values (Max-Min Computation). If the fused maximum value is below a threshold, it is inferred that snowfall will occur over the city. Similarly, MINMAX computation can be performed over the distributed WSN to detect the event of a fire.

The results obtained by simulating various protocols by varying the edges (i.e., coverage region of sensor nodes) and recording the scalar and vector files are shown below. Fig 6 and Fig. 7 are related to clustering. Because of the reduction in the number of effective sensors, the overall cost is reduced. Also, indirectly network lifetime is increased. Tables 1 and 2 are related to various grid sizes and simulation times and the effective cost of sensors in the field. If the sensor number doubles, the simulation time is also more. If the grid size increases, the effective cost with Algorithm 2 is less than Algorithm 1.

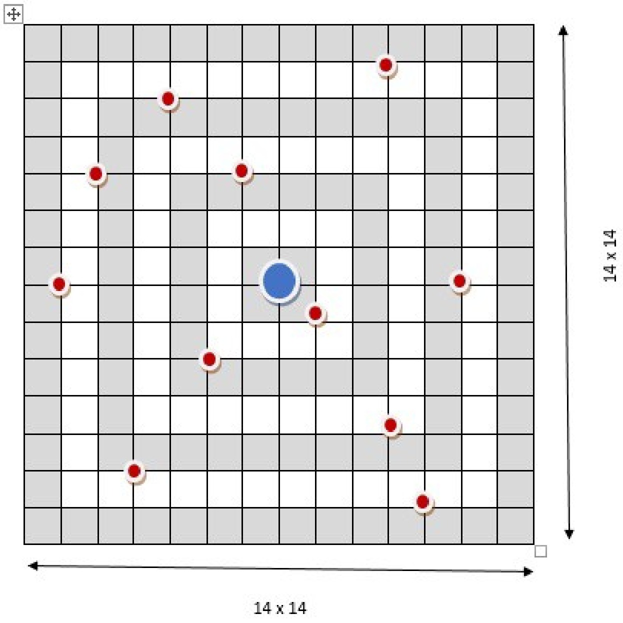
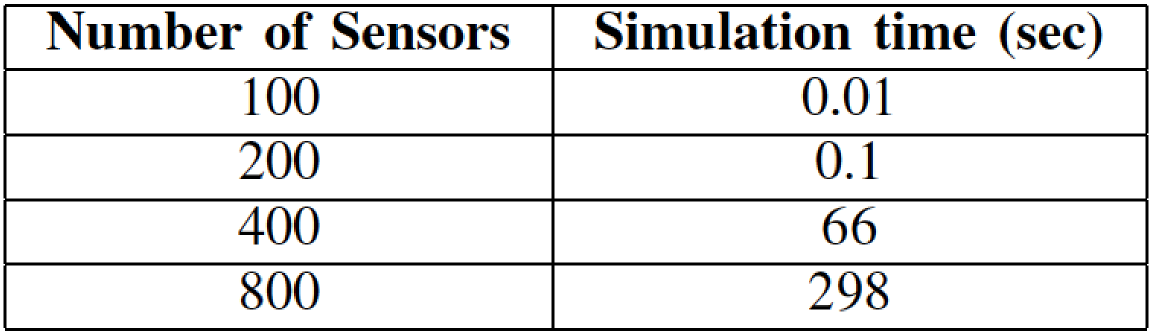


Fig. 5. Leveling with number of sensors in 14*14 WSN Matrix



Number of Sensors	Simulation time (sec)
100	0.01
200	0.1
400	66
800	298

TABLE I
NUMBER OF SENSORS WITH SIMULATION TIME FOR CLUSTERING AND DATA TRANSMISSION

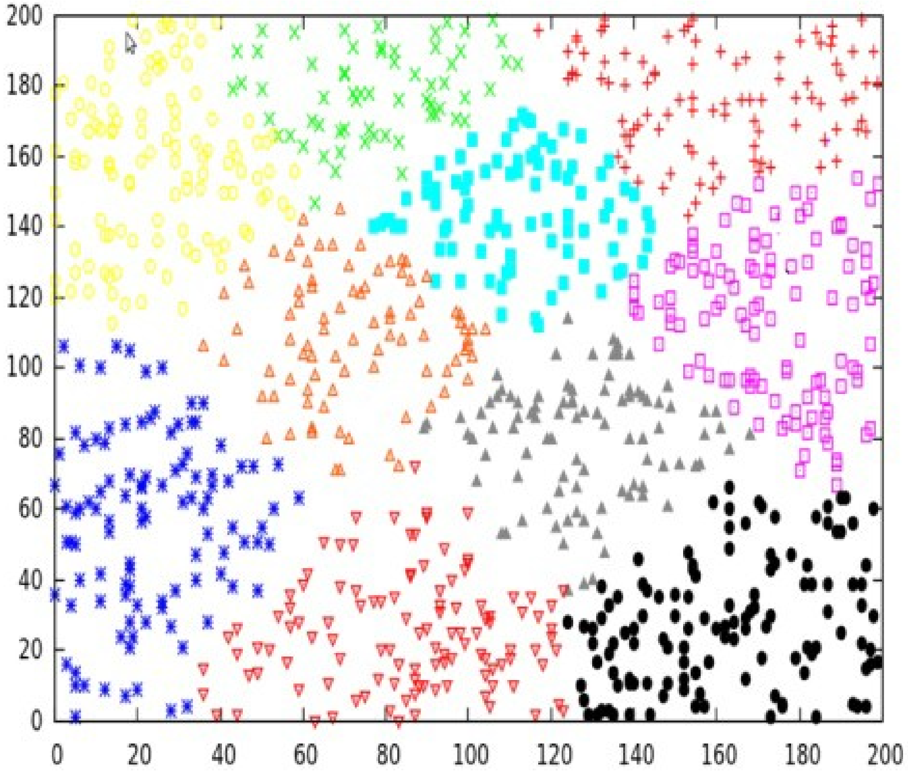


Fig. 6. 10*10 Matrix with uniform sensor field with clusters

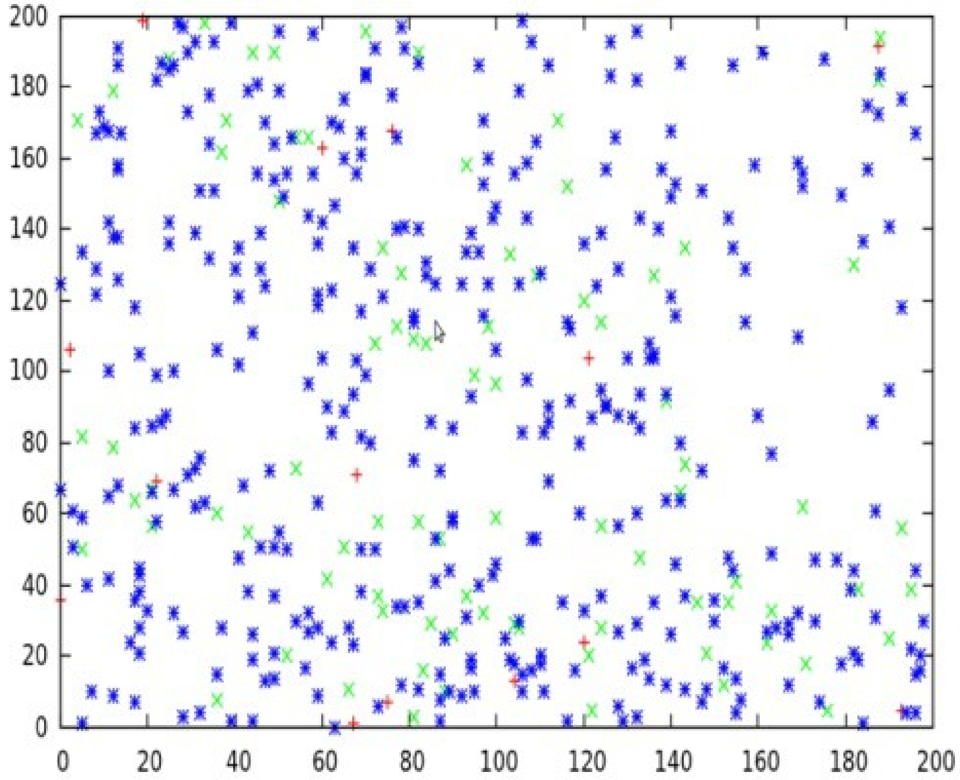


Fig. 7. 10*10 Matrix with sensor field after applying the cost minimized algorithms

Grid Size	Algorithm1 (Direct Method)	Algorithm2 (Divide and Conquer)
14*14	cost: 11200 Units	Using 4 7*7, 4*2700 = 10800 Units
20*20	cost: 13200 Units	Using 4 10*10, 4*3100 = 12400 Units
28*28	cost: 28400 Units	Using 4 14*14, 4*5700 = 22800 Units

TABLE II
GRID SIZE VS EFFECTIVE COST(ALGORITHM1 AND ALGORITHM2)

4.6 Conclusion

In this chapter, an interesting, combined cost as well as energy wastage minimization problem is formulated, and an exciting solution is proposed. Also, distributed computation and communication of minimum and maximum sensed value over a uniform rectangular grid are discussed. Event detection over a distributed wireless sensor network is also discussed. We expect these results to benefit real-world wireless sensor networks deployed over a uniform rectangular grid.

Chapter 5

Global Position Indexing

5.1 Introduction

This chapter addresses the problem of indexing physical locations across a planet (like time indexing based on dates/months/years) and provides an approach. Countries worldwide devised various geographical location specification codes (along with state/ province names, district names, city/ town/ village names, etc.). However, to this day, there has yet to be an addressing scheme to specify various types of geographical locations, such as hospitals, hotels, etc., worldwide. Apart from coarse grain localization of spatial locations using longitude, latitude (on the globe), and acceptable grain specification of spatial positions is an open research problem. This chapter provides an interesting approach to this problem and its applications. This is an effort in that direction to provide global position addressing. Various exciting applications of such local/global position indexing are provided. It is hoped that such indexing, when linked with other addresses such as IPv6 addresses, will be very useful.

Human civilization across the planet understood essential concepts associated with nature (physical reality): Space, Time, and Matter. Various calendars were proposed to specify the time, and various generations of gadgets were innovated to measure time. Also, the discovery that planets were spheres led to some ways of specifying the location of various geographical regions, such as local/global maps. Kepler discovered and utilized triangulation methods to discover the laws of planetary motion. The triangulation method formed the basis of the Global Positioning System (GPS) using satellites. Google deployed the technology enabling local navigation using Google Maps. Google also provides various navigational services on mobile phones, such as real-time voice-based navigational guidance. Sophisticated cars also provide interesting navigational aids.

5.2 Local Position Indexing: Node Representation and Application

The spatial locations on the planet are classified into various categories—for example, continents, oceans, forests, farms, etc. The invention of agriculture led to the widespread Occupation of land regions called farms. With the progress of civilization, land regions are designated for various purposes, such as houses, schools, temples, etc. Also, Homosapien was trained for various professions resulting in designations/classification of inhabitable regions into various categories such as hotels, administration places, forts, etc. The onset of the industrial revolution led to the birth of cities, towns, etc., and their occupation by various industries, hospitals, residential places, universities, etc. Thus, global spatial locations were locally classified/labeled and assigned different names in the local language. Also, time was organized using different calendars.

For navigation purposes, local and global maps were proposed. Advances in information & communication technologies and technologies like global positioning systems (GPS), the internet, World Wide Web enabled navigation using smartphones as a reality.

The planar (Euclidean plane) approximation of regions on the planet locally is good for navigation using cars, trucks, trains, and other such vehicles. Thus, the locations on the plane can be well represented using the graph data structure. The vertices of a weighted graph (of different types of weights such as distance, road conditions, etc.) belong to different types/classes, such as hospitals, hotels, schools, universities, prisons, etc.

5.2.1 Goal

To find a simple representation for nodes/ vertices of a graph locally and globally such that every spatial location on the planet is uniquely specified with respect to type/class and other attributes (e.g., country, state, district, etc.).
This objective relates to the utilization of IPv6 addresses in the realization of IoT. We propose utilizing some of the bytes in 16-byte IPv6 addresses to specification various types of nodes on the planet.

5.2.2 Tuple idea

There are 195 countries worldwide. These countries have a number of states, cities, towns, villages, etc. Further, each of these cities/towns has a number of various institutions like hospitals, universities, restaurants, etc.

Let us say a tourist wants to take a tour of a city. Now, this person needs to know about various tourist spots, hotels (where he wants to stay), railway stations (from where he can catch a train), and other such places. Many institutions will have further constraints like rating, comfortability, good services, near railway stations or airports, and cost efficiency. Since he is new to the city, he will need help making decisions.

Therefore, we need to think of a way to index every building of an institution considering all the mentioned and other unmentioned constraints. Not only indexing, but we also need to think of a way through which we can represent these institutions and buildings as nodes of graphs which then can be optimized using suitable graph optimization algorithms. This will, in turn, help that particular person in making the best possible decisions. There can be various kinds of institutions in a particular city. Some of these are as follows:

1. Hospitals
2. Police station
3. Railway station
4. Restaurants
5. Hotels
6. Airport
7. Malls
8. Movie theatre
9. Various kinds of shops
10. Educational institutions (schools, colleges, etc.)
11. Administrative buildings
12. Gardens and parks (stadiums)
13. Bus stop
14. Temples
15. Club and bars

All these institutions can have various constraints, as discussed earlier.

5.2.3 Tuple idea (for indexing)

Like an IP address, we can assign every institution a unique tuple. For now, this tuple consists of 5 elements. Element 1 is assigned to a particular country. E.g., if The code for India is 91, then we can assign element one as 91 to represent India. Elements 2 and 3 are assigned to representing a state and city in that country (which is represented by element 1), respectively. Element 4 is assigned to an institution like a hospital, temple, hotel, etc. Element 5 will be assigned value according to the various constraints on institutions represented by element 4.

This way, we can assign a unique tuple to an institute with a particular set of constraints/requirements. In other words, a unique tuple is used for indexing it based on different attributes/types associated with a geographical location. We expect many applications for global position indexing using unique tuples. We explicitly specify some applications in the following discussion.

1. Apartment / House selection: Recommended communities in a city
2. Hospitalization satisfying various constraints (cost, nearness to a hotel/restaurant, etc.)
3. Hotel reservation (for rent, for offering a party, etc.)
4. Planning for future expansion of a city (gated communities, business parks, etc.).

In operations research and other related disciplines, several interesting problems are already well-studied, and some solutions are already proposed. Some of them are:

1. Transportation and Trans-shipment Problems
2. Travelling Salesman Problem
3. Community Extraction in Social Networks
4. Maximum Flow Problem
5. All-pairs Shortest Path Problem
6. Graph Coloring Problems

We expect graph/hypergraph-based optimization problems to naturally arise in navigation based on local/global position indexing. Some of these problems involve the selection of nodes, paths, subgraphs, cuts, spanning trees, Hamiltonian tours, independent vertex sets, etc.

In navigation problems, typically, one needs information related to details associated with various types of nodes (cost of hospitalization, etc.). Also, nodes of various types need to be visited based on optimal attributes related to the nodes. Also, one needs to locate a sub-graph/ sub-hypergraph in the region (e.g., city) being navigated (e.g., need a hospital close to a hotel, medical store, or restaurant, which is cheaper than available ones). More generally, graph/hypergraph-based optimization problems naturally arise with various adjacency/nearness constraints.

Thus, in summary, known graph/ hypergraph theoretic concepts, problems, and algorithms are helpful (in the global position indexing-based navigation). However, more interestingly, novel graph/hypergraph concepts, problems (optimization, sorting, searching, etc. problems), and algorithms will naturally arise and are needed to be understood/ discovered.

In this section, we mainly discussed local position indexing. We briefly pointed out global position indexing with tuple ideas. We now look closer at issues and applications of position indexing globally.

5.3 Global Position Indexing: Applications

The invention of the telephone led to the first communication network, a wired network. It led to the concept of addressing the nodes of a communication network, namely the global telephone numbers. Over time, various communication networks (Ethernet, Internet) needed IP addresses, MAC addresses, etc. To be able to specify the address in computers/digital circuits, the addresses are proposed to be a certain number of bytes (bits). In the case of IP addresses, they are also linked to a Domain Name System (DNS) to enable straightforward mnemonic-based Representation.

In some sense, the Tuple idea for global position addressing is a natural representation for specifying a physical location globally. We are naturally led to the following questions:

1. How large should the global location address be given future demands? (The answer to this question needs known information about global dimensions (say the inhabited area/volume, forest area, ocean area/volume, etc.). For instance, with residential towers, one needs to consider that there can be multiple floors in a building. In some sense, divide the space into fine-grain locations (to the resolution desired, e.g., floors in a building) and index them into various types based on the attributes. Most generally, we are led to the need for SPACE-TIME INDEXING schemes.)
2. Can such an addressing scheme be updated based on future demands? Should IPv6 addresses (16 bytes) be linked/ Associated with tuple-based location addresses? (i.e., Linking the Internet of Things with the Physical Location of various types of institutions that are present on the planet) (Note: As for Internet addresses, Domain Name System type binding for tuples associated with spatial location can be facilitated (e.g., For a location INDIA, AP, Guntur,…..). It should be noted that in the case of IP addresses, the tuple value cannot be fixed to 256. However, such restriction may not be possible with position-related tuples.)

Applications:

1. Tourism Across World: Global location indexing enables OPTIMAL PLANNING (in terms of cost, comfort, time-efficiency, etc.) of locations to be visited in some time sequence, in some time frame, etc.
2. For military and civilian applications, navigation of DRIVERLESS vehicles can be facilitated using global indexing.
3. Intelligent Transportation Systems can be enabled for monitoring the position of vehicles rendering issues like accident notification, congestion status, etc. Also, global position indexing could enable terrorist plot forecasting and crime prediction.
4. City Administration: Most locations in a city anywhere in the world are associated with the following numbers: (a) landline and cell phone numbers, (b) Aadhar/Social Security numbers, (c) local IP addresses, (d) Fax numbers, (e) Barcodes, (f) Ethernet Card codes, (g) ZIP/PIN codes, (h) Airport/Train Station codes, (i) State/Country/District, etc. codes, etc. It will be very convenient for administrative officials to relate these with tuple-based indexing associated with physical locations (such as apartments, hotels, etc.). Real-Time Navigation with various types of constraints globally for civilian and military applications.

5.4 Conclusion and future work

This chapter considers the problem of indexing various physical locations on the planet. Various interesting graph/ hypergraph problems naturally arise on a local position scale. For some such problems, graph algorithms are known. For future work, we need to devise a way (maybe a formula) to assign weight to any institution and the path (edges) between any two institutions based on their corresponding tuple. This will give us a concise graph problem with edges and weights. We can then work on various optimization algorithms for these problems, ultimately giving us the best possible solution. E.g., this will give our tourists the best way to tour the city (where the best way means devising a tour considering his/her requirements and constraints).

Chapter 6

Conclusion and future work

In this thesis, we proposed new LEACH Protocols which are better than other discussed LEACH protocols in many ways. We have proposed the LEACH-WC protocol, which distributes the energy between the nodes in the network as an efficient way to reduce energy wastage from a global perspective and increase system lifetime. We have also formulated an exciting problem on combined cost and energy wastage minimization and proposed an interesting solution. We have also discussed distributed computation and communication of minimum and maximum sensed value over a uniform rectangular grid. Event detection over a distributed wireless sensor network is also discussed. We also proposed an approach for indexing various physical locations on the planet.

Regarding future work, we need to consider different types of deployment and mobility of the nodes and compare them with the existing mobility-based routing protocols to implement the LEACH-WC protocol in MWSN. We must also look into how the proposed work can have real-world utility. We also need to combine the proposed work with the indexing system to formulate a problem and then find its solution, which can lead to an intelligent transportation system.

Chapter 7

Bibliography

1. Wendi Rabiner Heinzelman, Anantha Ch, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks, pages 3005{3014, 2000.
2. Sarbjeet Kaur, Ravi Gupta, “Deploying an Optimized LEACH-C Protocol for Wireless Sensor Network,” International Journal of Advanced Research in Computer Science and Software Engineering, vol.5, issue 7, July 2015.
3. J. Xu, N. Jin, X. Lou, T. Peng, Q. Zhou and Y. Chen, "Improvement of LEACH protocol for WSN," 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 2012, pp. 2174-2177, doi: 10.1109/FSKD.2012.6233907.
4. Gaba, Gurjot & Monga, Himanshu & Kansal, Lavish & Miglani, Rajan. (2016). I-LEACH Protocol for Wireless Sensor Networks. International Journal of Multimedia and Ubiquitous Engineering. 11. 279-292.
5. Al-Baz, Ahmed & El-Sayed, Ayman. (2015). Energy-Aware Enhancement of LEACH Protocol in Wireless Sensor Network. 10.1109/ICCTA37466.2015.9513448.
6. Sinde, Ramadhani & Begum, Feroza & Njau, Karoli & Mollel, Michael. (2020). Lifetime improved WSN using enhanced-LEACH and angle sector-based energy-aware TDMA scheduling. Cogent Engineering. 7. 10.1080/23311916.2020.1795049.
7. X. Wang, J. Zhao, J. Hou, X. Tang, H. Wu and P. He, "Research on WSN Topology Algorithm Based on Greedy Shortest Paths," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 5215-5220, doi: 10.23919/CCC50068.2020.9189355.
8. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, ”Wireless sensor networks: A survey,” Comput. Netw., vol. 38, no. 4, pp. 393-422, 2002.
9. Shen, J., Wang, A., Wang, C., Hung, P.C., and Lai, C.F., ”An efficient centroid-based routing protocol for energy management in WSN-assisted IoT. Ieee Access, 5, pp.18469-18479, 2017.
10. T. Jagannadha Swamy, G. Ramamurthy, and P. Nayak, ”Optimal, Secure Cluster Head Placement Through Source Coding Techniques in Wireless Sensor Networks,” in IEEE Communications Letters, vol. 24, no. 2, pp. 443-446, Feb. 2020, doi: 10.1109/LCOMM.2019.2953850.
11. Mohammed Nazeer 1 *, Garimella Rama Murthy 2, “Protocols in mobile cognitive wireless sensor networks: A survey” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 pp. 10268-10275 (2018).
12. Xiaodong Xian, Weiren Shi and He Huang “Comparison of OMNET++ and Other Simulator for WSN Simulation”College of Automation, Chongqing University , Chongqing, 400044, China Email: xxd@ccee.cqu.edu.cn,wrs@cqu.edu.cn,

dick_hh@163.com 978-1-4244-1718-6/08/$25.00 ©2008 IEEE PP:1439-1443.

13. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, ”Wireless sensor networks: A survey”, Comput. Netw., vol. 38, no. 4, pp. 393-422, 2002.
14. Shen, J., Wang, A., Wang, C., Hung, P.C. and Lai, C.F., ”An efficient centroid-based routing protocol for energy management in WSN-assisted IoT. Ieee Access, 5, pp.18469-18479, 2017.
15. Tata Jagannadha Swamy, Jayant Vaibhav Srivastava, and Garimella Ramamurthy. ”Non Uniform Grid Based Cost Minimization and Routing in Wireless Sensor Networks.” International Journal of Wireless Networks and Broadband Technologies (IJWNBT) Vol.2, no . 1,pp.16-28, 2012.
16. Tata, Jagannadha Swamy, Garimella Ramamurthy, and Kvsvr Prasad. ”Novel Energy Efficient Dynamic Routing Protocol for Wireless Sensor Networks.” International Journal of Advanced Research in Computer Science 5, no. 7 (2014).
17. Guleria, Kalpna, and Anil Kumar Verma. ”An energy efficient load balanced cluster-based routing using ant colony optimization for WSN.” International Journal of Pervasive Computing and Communications Vol.3, no.4, pp. 233-246, 2018.
18. T. J. Swamy, T. Sandhya and G. Ramamurthy, ”Link evaluation of uniform gridbased wireless sensor networks to base station with leveling and clustering,” 2013 ICCCI, 2013, pp.1-5, doi: 10.1109/ICCCI.2013.6466146.
19. R. Akl, P. Kadiyala and M. Haidar, ”Non-uniform grid-based routing in sensor networks,” 2009 IEEE 9th Malaysia International Conference on Communications (MICC), Kuala Lumpur, 2009,pp. 536-540, doi:10.1109/MICC.2009.5431440.
20. Dr.G.Ramamurthy, MD Nazeer, Dr. Tata Jagannadha Swamy, ”Energy Efficient Clustering in Real World Wireless Sensor Networks : Implementation”,ICSCSP-2020, Vol.1, pp.1-6,2020
21. Vinodhini, R. and Gomathy, C., ”MOMHR: A Dynamic Multi-hop Routing Protocol for WSN Using Heuristic Based Multi-objective Function”, Wireless Personal Communications, Vol.111, no.2, pp.883-907, 2020.
22. Thekkil, Tibin Mathew, and N. Prabakaran. ”A comparative analysis of quality parameters in wireless sensor network with different techniques: a survey.” International Journal of Mobile Network Design and Innovation, Vol.8, no. 3,pp.141-156, 2018.
23. Swamy, Tata Jagannadha, and Garimella Rama Murthy. ”Cost Minimization of Sensor Placement and Routing in Wireless Sensor Networks: Placement and Routing Issues in a Random Plane.” In Sensor Technology: Concepts, Methodologies, Tools, and Applications, pp. 1286-1301. IGI Global, 2020.