LIFETIME ENHANCEMENT OF WIRELESS SENSOR NETWORK WITH DATA AGGREGATION

A Thesis Submitted in Fulfillment of the Requirements for the Award of the Degree of

MASTER OF ENGINEERING
in
Wireless Communication

Submitted by

MANDEEP KAUR
Roll No: 801663004

Under Supervision of
Dr. Amit Munjal
Assistant Professor, ECED

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB
June 2018

DECLARATION

I Mandeep Kaur hereby declare that the work presented in the thesis entitled "**Lifetime Enhancement of Wireless Sensor Network with Data Aggregation**", in fulfillment of the requirement for the award of the degree of Master of Engineering (Wireless Communication) submitted at Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of **Dr. Amit Munjal** (Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology) from June 2017 to June 2018. The matter presented in this thesis has not been submitted either in part or full to any other university or institute for the award of any other degree.

Date: 06/08/2018

Mandeep Kaur
06/08/2018

Mandeep Kaur
801663004

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date: 06/08/18

06/08/18
Dr. Amit Munjal
Assistant Professor
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology
(Deemed to be University), Patiala

ACKNOWLEDGEMENT

First, I would like to express my sincere gratitude to my supervisor **Dr. Amit Munjal** for his continuous support at every step of my ME program. His guidance helped me at all the time in my research period. And also he guided me the lessons of life “set your targets in life” and motivated me at every step. Without his unfailing support and belief in me, this thesis would not have been possible. And for this, I am truly grateful.

I also like to express my gratitude to **Dr. Alpana Agarwal** Head of ECED Department for his constant motivation and encouragement. And would also like to thanks to **Dr. Ashutosh Kumar Singh** PG coordinator for his support.

I am thankful to all the faculty members of ECED Department for their positive and co-operative response with time, energy and valuable suggestions they gave me to fulfill the task. I would like to thanks my friends who helped me at every step and from whom I have learnt the art of happiness and never giving up approach.

Finally, I would like to express my sincere and deep gratitude to my parents and family members for their love, encouragement, care and support. At every step, without their faith and support on me, I could not complete my ME program.

Mandeep Kaur

ABSTRACT

Nowadays wireless sensor networks (WSN) are widely used in various domains such as disaster relief operation, military applications, environmental monitoring, home applications, structural monitoring, agricultural sector, health applications etc. Energy saving is the rudimentary provocation in wireless sensor networks. The energy of the WSNs can be conserved in many ways such as: duty cycling of nodes, clustering, energy efficient routing, data energy etc. The data aggregation is one of the approach which is used to conserve the energy of the WSN nodes by removing redundant packets from the collected packets. This helps in increasing the network lifetime and on the other hand, it saves the network bandwidth as well. Data aggregation algorithms are broadly classified on the basis of their data collection strategy into three types. In this thesis we have attempted to present a broader review of the data aggregation algorithms on the basis of their categorization as discussed earlier. Moreover, we have also proposed a centralized data aggregation algorithm that is Levelling and Self Localization based Clustering(LSLC) algorithm. This algorithm allows the nodes to self-localized themselves and also the data aggregation is performed at multiple levels which reduces the energy consumption of the nodes. This also reduces the number of packets sent to the base station and thereby enhances the network lifetime. In addition to this, the LSLC algorithm is compared with different existing data aggregation algorithms in terms of network lifetime, residual energy, packet to delivery ratio, latency, packet sent to the base station. Initially, the simulation results are obtained to analyze the impact of different metrics like increasing the number of nodes in the same area, with and without data aggregation etc. Then the LSLC is compared with some existing algorithm and it shows a considerable improvement in network lifetime, residual energy etc.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATIONS

E	Energy cost
C	Incremental energy cost
t_p	Expiry of timer
T	Threshold
S	Source node id
P	Previous hop node id
D_{is}	Hop count from the node i to the source nodes
h_j	Hop count of the node j to the sink nodes
h_i	Hop count from the node i to sink node
h	Hop distance
t	Number of bits
d_{is}	Distance between transmitter and receiver
d_o	Communication radius of the node
E_o	Initial energy
E_{elec}	Energy consumption of circuit per bit by transmitter and receiver
E_{fs}	Energy consumption of amplifier in communication range
E_{amp}	Energy consumption of amplifier when distance is increased from its communication range.
E_{DA}	Data aggregation energy
P_1	Power levels
P_r	Power received by the receiver
P_t	Power transmitted by the transmitter
G_r	Gain of the receiver antenna
G_t	Gain of the transmitting antenna
h_t	Height of the transmitting antenna
h_r	Height of the receiving antenna
d	Distance between transmitter and receiver
f	Distance between base station and Node 1
e	Distance between Node 1 and Node 2
x_{-} B.S	x co-ordinate of base station
y_{-} B.S	y co-ordinate of base station

x_{-} Node 2	x co-ordinate of Node 2
y_{-} Node 2	y co-ordinate of Node 2
y_{-} Node 1	y co-ordinate of Node 1
x_{-} Node 1	x co-ordinate of Node 1
N	Number of nodes
c	Number of clusters
N/c	Number of nodes per cluster
x	Number of cluster head

LIST OF ABBREVIATIONS

BECDA	Bandwidth efficient cluster based aggregation
BS	Base Station
BTF	Balanced tree formation
CCMAR	Cluster-chain mobile agent routing
CDA	Cooperative data aggregation
CDAS	Cluster based DAS
CEA	Centralized energy allocation
CESPT	Confusion based energy-saving and privacypreserving data aggregation
CH	Cluster head
CMCT	Centralized maximum cover tree
CS	Compressive sensing
DAHDA	Dynamic adaptive hierarchical data aggregation
DAS	Distributed data aggregation scheduling
DC	Density based clustering
DD	Direct diffusion
DEA	Distributed energy allocation
DEEC	Distributed energy efficient clustering scheme
DMCT	Distributed maximum cover tree
DSG	Directional source grouping
EAMMH	Energy aware multihop multipath hierarchical
EDAHDA	Extended DAHDA
EECA	Energy efficient clustering algorithm
EECDA	Energy efficient clustering and data aggregation
EECF	Energy efficient collision free
EECP	Energy efficient cluster-chain based protocol
EECS	Energy efficient clustering scheme
EED	End to end delay
EEHA	Energy-Efficient and High Accuracy
EESAA	Energy efficient sleep awake aware
EHC	Extended hop count
GCF	Global closest first

GHS	Gallagher-Humblet-Spira
GIT	Greedy Incremental tree
GPS	Global positioning system
GT	Greedy transmission
HEED	Hybrid energy-efficient distributed clustering
IEEE	Institute of Electrical and Electronics Engineers
IEMF	Itinerary energy minimum for first-source-selection
ILS	Iterated local search
LAs	Local aggregators
LEACH	Low energy adaptive clustering hierarchy
LOS	Line of sight
LSLC	Levelling and Self-Localization based Clustering
MA	Mobile agents
MAAD	Mobile agent-based directed diffusion
MANET	Mobile ad hoc network
MAPA	Mobile agent protocol based energy aware data aggregation
MAs	Master aggregators
MHC-DC	Meta-Heuristic computational-density based clustering
MODLEACH	Modified -low energy adaptive clustering hierarchy
NACK	Negative acknowledgement
NLOS	Non-line of sight
NOID	Near-optimal itinerary design
OCCN	Optimal clustering in circular networks
OPD	Optimized power declaration
PCA	Principal component analysis
PDA	Peony tree based data aggregation
PDR	Packet to delivery ratio
PEDAP	Power efficient data gathering and aggregation protocol
PEDAP-PA	Power efficient data gathering and aggregation protocol-power aware
PPSDA	Privacy preserving secure in-network data aggregation
RAG	Real time data AGgregation
RSS	Received signal strength
RSSI	Received signal strength indicator
SCPDA	Secret confusion of privacy-preserving data aggregation

SDAP	Secure data aggregation protocol
SeDC	Secure data collection scheme based on compressive sensing
SEEDA	Secure end-to-end data aggregation
SEP	Stable election protocol
SLMADA	Scalable and load balanced mobile-agents based data aggregation
SMART	Slice-Mix-AggRegaTe
SN	Selected node
SP-DAS	Shortest path DAS
SPEED	Stateless Protocol for Real-Time Communication
SQL	Structured query language
TAG	Tiny AGgregation
TBID	Tree based Itinerary design
TDMA	Time division multiple access
TTCDA	Two-tier cluster based data aggregation
TTD	Time to deadline
TTD-EED	Time to deadline-end to end delay
TTL	Time to live
WSN	Wireless sensor network

CHAPTER 1

INTRODUCTION

The networks are generally categorized as wired or wireless, the wired networks offer more complexity due to increase in the number of connecting wires. So, most of the networks are nowadays wireless. Wireless networks are classified into types: Infrastructure based and infrastructure less. Infrastructure based wireless network are those networks in which access points make the connection with a wired network and provide the connection to the wireless devices. The example of such kind of networks is IEEE 802.11 LAN (Local area network). On the other hand, infrastructure less wireless network, are those networks in which no access points are required. The devices can communicate with each other whenever they come in contact with other devices within their radio range. The example of such types of networks is MANET (Mobile Ad Hoc Network), Wireless Sensor Network (WSN) etc. The Wireless sensor network is an infrastructure-less network which is small in size, cost-effective and task-specific sensing network. WSN is deployed in the area where humans cannot reach every time but the information pertaining to that area is required all the time. The WSN consist of nodes that sense the information from the particular area and communicate same to the base station by a single hop or multihop. The node is the basic element of WSN which consists of a processor, battery, Rx-Tx antenna, memory, specific sensors etc. [1].

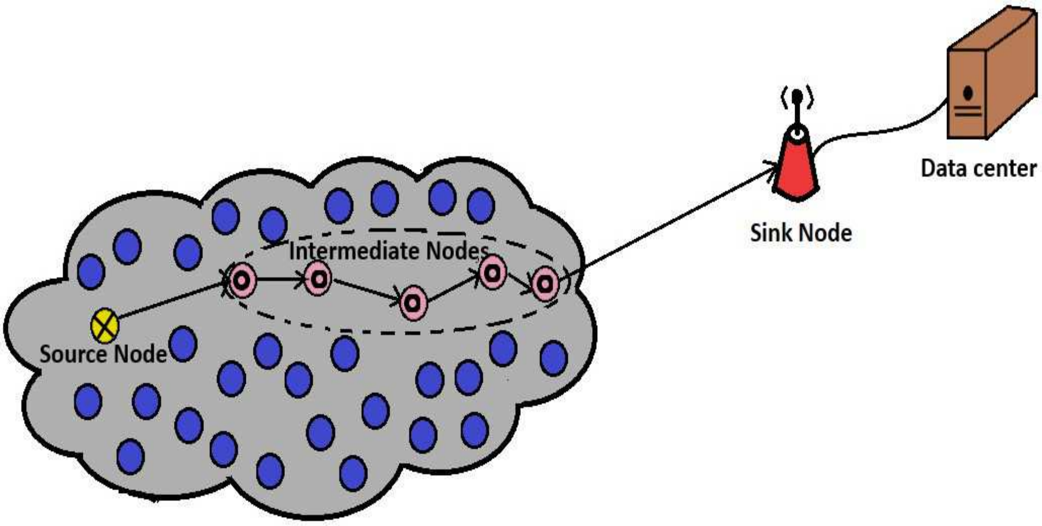


Figure 1.1: General architecture of Wireless Sensor Network

The general architecture of wireless sensor network consist of three different types of node as shown in Figure 1.1 and are discussed below:

- **Source node:** It will collect the required amount of information from the environment in which it is deployed.
- **Intermediate node:** Nodes that forward the information sent by the source node toward the destination node.
- **Sink node:** It is a base station node which collect the information from the source node through intermediate nodes.

There are various applications of wireless sensor networks like, disaster relief operation, military applications, environmental monitoring, home applications, structural monitoring, agricultural sector, transportation, health applications [1, 2] as presented in Figure 1.2. In several applications of WSNs, different sensor nodes are prone to transmit redundant or correlated information towards the sink, which leads to the wastage of energy as well as network bandwidth. This overall results in the wastage of network capacity and thereby reduces the network lifetime. Thus there is a dire need to overcome this problem. One of the possible solution to overcome these problems is to use data aggregation algorithms. These data aggregation algorithms not remove the redundant information but it also saves the network energy as well as network bandwidth [4].

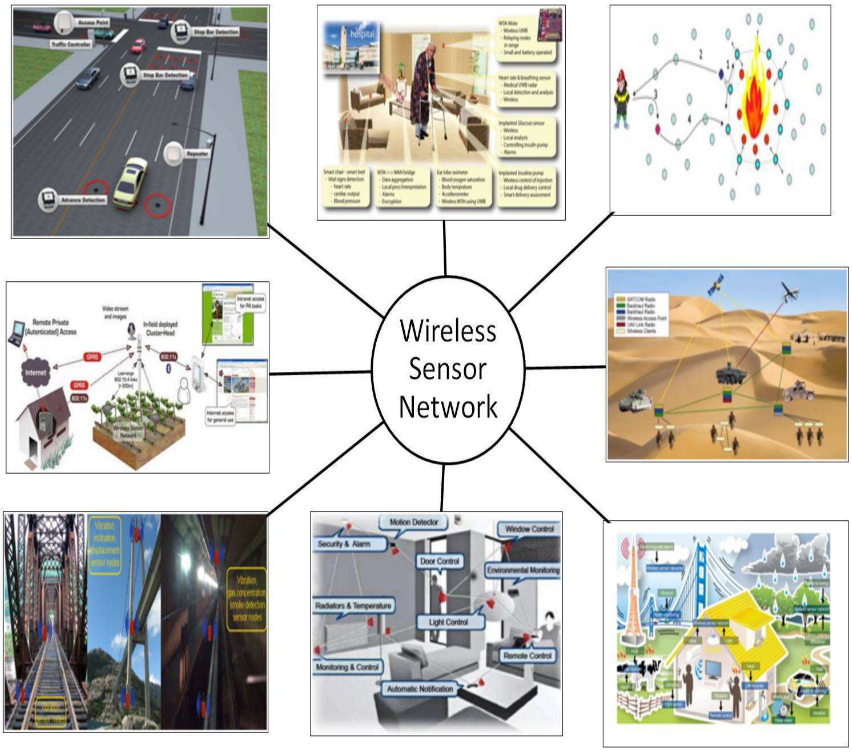


Figure 1.2: Applications of Wireless Sensor Network

1.1 DATA AGGREGATION

The main aim of deploying the wireless sensor network in a specific area is to collect the information from that area and it is termed as data gathering [4]. The gathered information contains a lot of redundant information and if this information is forwarded as it is to the sink node, then it leads to more energy consumption by the nodes. Thus, we need to extract the useful data from the collected information with the help of data aggregation algorithm. This can be explained with the help of an example in Figure 1.3. Let's suppose we have 7 sensor nodes which are grouped in two smaller clusters as in Figure 1.3 out of which two nodes are reserved as data aggregator nodes or cluster head nodes (say as Node6, Node7). The responsibility of cluster head (CH) nodes is to gather the information from all the sensor node within its cluster and then perform some data aggregation function such as MAX function [28] is given in Equation (1.1) and (1.2). The Table 1.1 shows the information received by 5 sensor nodes for temperature and pressure respectively. The CH will send the MAX information as computed using Equation (1.1) and (1.2) to the base station. This saves the energy of nodes and hence maximizes the lifetime of WSN. Moreover, the data aggregation also saves the bandwidth requirement of the channel. Apart from the above advantages, the data aggregation comes with some disadvantage as well. The major disadvantage is latency. As the CH performs different levels of data aggregation so it increases the latency of the overall network.

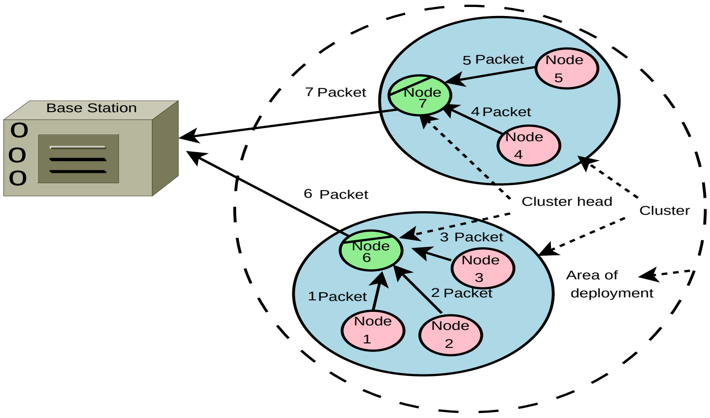


Figure 1.3: Data aggregation example

Information sent by data aggregator (node6) is,

$$MAX(Pressure) = MAX(150, 100.2, 200) = 200$$

$$MAX(Temp.) = MAX(7.4, 3.2, 10) = 10 \tag{1.1}$$

Table 1.1: Information accumulate by nodes

S. No	**Node ID**	**Packet ID**	**Pressure (kPa)**	**Temp.** (C^o)
1	Node1	Packet1	150	7.4
2	Node2	Packet2	100.2	3.2
3	Node3	Packet3	200	10
4	Node4	Packet4	98.4	2
5	Node5	Packet5	100.7	4.2

Information sent by data aggregator (node7) is,

$$\begin{gathered} MAX(Pressure) = MAX(98.4, 100.7) = 100.7 \\ MAX(Temp.) = MAX(2, 4.2) = 4.2 \end{gathered} \tag{1.2}$$

The different approaches used for data aggregation are, tree based approach, cluster based approach, and multiple path based approach etc. Data aggregation is used to enhance the network lifetime, reduce the network overhead, and reduces the bandwidth of the wireless network.

1.2 BROAD CLASSIFICATION OF DATA AGGREGATION ALGORITHMS

The data aggregation algorithms are broadly classified into three types as shown in Figure 1.4

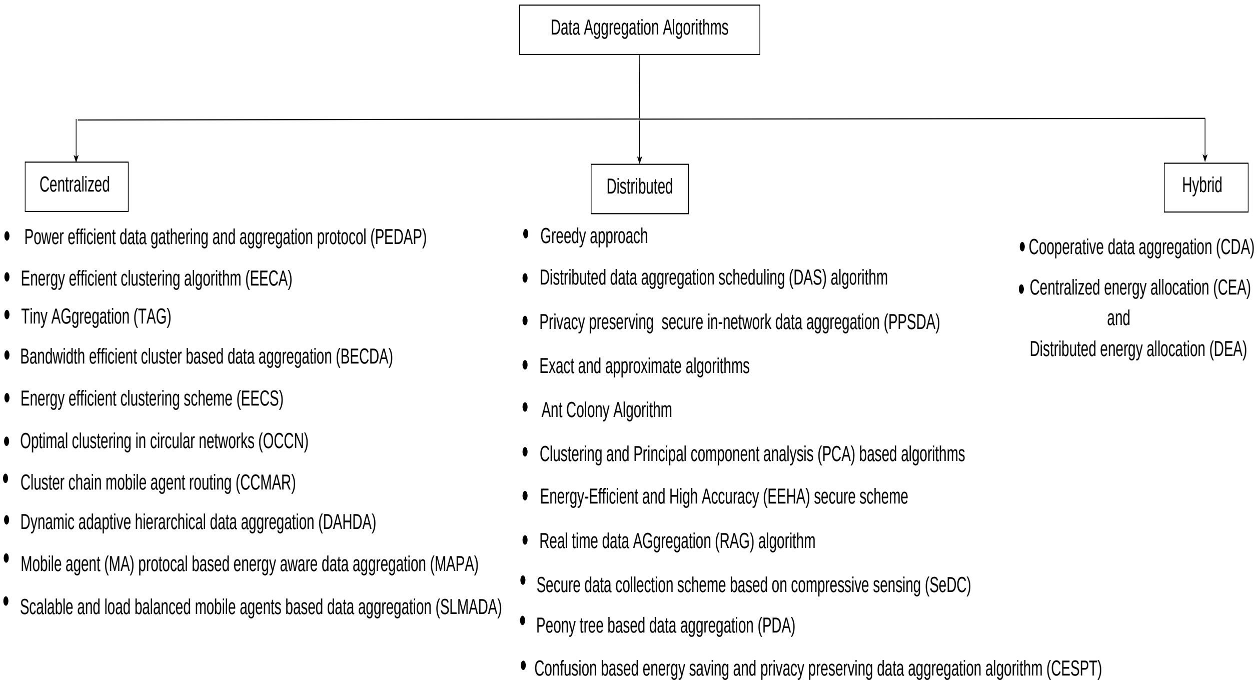


Figure 1.4: Data aggregation algorithms

1. **Centralized Algorithm:** In these algorithms there are some central nodes which are responsible for maintaining the information about all other nodes in the network as shown in Figure 1.5. These central nodes will collect and aggregate the data and send it to the base station as and when required. Some of the existing centralized data aggregation algorithms are: Power efficient data gathering and aggregation protocol(PEDAP) algorithm [6], Energy efficient clustering algorithm (EECA) [7], Tiny AGgregation (TAG) approach [8], Bandwidth efficient cluster based Aggregation (BECDA) algorithm [9], Energy efficient clustering scheme(EECS) [10], Optimal clustering in circular networks (OCCN) algorithm [11], Cluster-chain mobile agent routing (CCMAR) algorithm [12], Dynamic adaptive hierarchical data aggregation (DAHDA) algorithm [13], Mobile Agent protocol based energy aware data aggregation (MAPA) [14], Scalable and load balanced mobile-agents based data aggregation (SLMADA) [15].

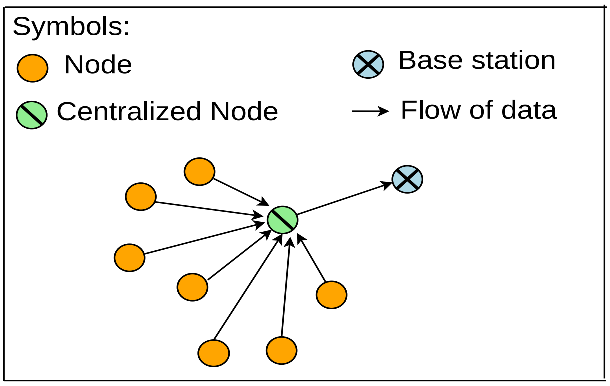


Figure 1.5: Centralized algorithm

2. **Distributed Algorithm:** In distributed data aggregation algorithm there is no central node which is responsible for gathering and aggregating the information as shown in Figure 1.6. All the nodes are equally responsible for routing their packets towards the base station. Some of the existing distributed data aggregation algorithm are: Greedy approach

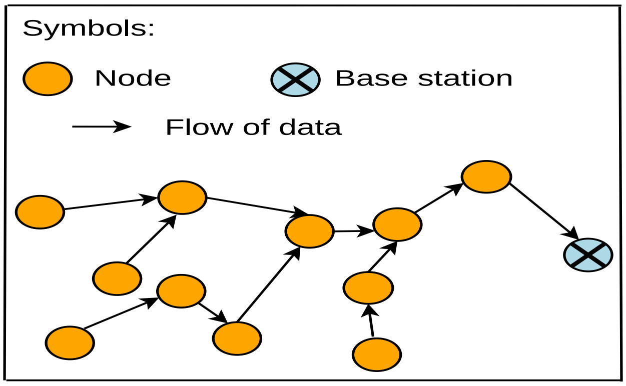


Figure 1.6: Distributed algorithm

[16], Distributed data aggregation scheduling (DAS) algorithm [17], Privacy preserving secure in-network data aggregation (PPSDA) algorithm [18], Exact and approximate algorithms [19], Ant Colony Algorithm [20], Clustering and Principal component analysis (PCA) based algorithms [21], Energy-Efficient and High Accuracy (EEHA) secure scheme [22], Real time data AGgregation (RAG) algorithm [23], Secure data collection scheme based on compressive sensing (SeDC) [24], Peony tree based data aggregation (PDA) [25], Confusion based energy-saving and privacy-preserving data aggregation (CESPT) algorithm [26].

3. **Hybrid Algorithm:** The hybrid algorithm is a combination of both centralized and distributed algorithm. Some of the existing hybrid algorithm for data aggregation are: Cooperative data aggregation (CDA) Algorithms[27], Centralized energy allocation (CEA) and Distributed energy allocation (DEA) algorithm [28].

1.3 MOTIVATION OF DATA AGGREGATION

In WSN each node consumes energy in three different domains i.e. in communication, sensing, and processing. The communication energy of a node is variable and it depends upon the distance involved in communication while the sensing and processing energy are fixed and it depends upon the type of manufacturer of WSN node. Thus in order to enhance the lifetime of WSN, we need to reduce the communication energy with the help of energy efficient algorithm, data aggregation etc. The data aggregation is a process by which the redundant packets are removed and thus the energy required for sending these redundant packets is saved. The data aggregation conservers not only the energy of the network but it also saves the network bandwidth and packet delivery ratio. This motivates to pursue our research in data aggregation domain.

1.4 CHALLENGES OF DATA AGGREGATION

There are different types of challenges in data aggregation. Some of them are discussed as follows:

- **Aggregation function:** It is a function that is used to reduce the number of packets sent to the base station. There are different data aggregation functions that can be used as discussed above. Reduction in the number of packets depending upon the type of data aggregation function being used. The selection of aggregation function that minimizes the bandwidth as well as maximizes the lifetime is one of the challenging tasks.

- **Latency:** When the packets are aggregated at different points within a network, then the aggregator node takes some time to aggregate the data. This increases the time taken by a packet to reach at the destination node. Thus it becomes challenging to divide the number of aggregator nodes that a packet need to travel so that latency should be minimum.

- **Number of Data Aggregator node:** The number of data aggregator nodes plays an important role in data aggregation. Higher the number of data aggregator node, higher is level of data aggregation. This results in a reduction of a number of packets sent and thus the energy of the nodes is saved. On the other hand, latency will be increased. Thus there exist a tradeoff between the number of data aggregation nodes and latency.

- **Selection of Data Aggregator node:** One of the major problem faced during data aggregation is the selection of data aggregator node from the network. The data aggregator node must be selected on the basis of various evaluation metrics such as residual energy, node degree etc. so that network lifetime can be enhanced.

- **Energy consumption:** In WSN, each node consumes energy in their main tasks i.e. sensing, processing and communication. The amount of energy consumption in communication is far more than sensing and processing. The communication energy depends upon the distance between transmitter and receiver. So we need to select the transmission distance in such a way that minimizes the energy consumption.

- **Data accuracy:** When different data packets are aggregated, then there is a high probability that some packets may be lost and this results in the error in the output. Thus to maintain the data accuracy while performing data aggregation is also one of the difficult tasks.

1.5 IMPORTANCE OF DATA AGGREGATION

Consider a scenario in which the energy of a node is very less and it needs to send the gathered information then it is not possible for that node to transmit the same. This is because data gathering (is a process through which all the sensed data from the environment is sent as it is to the base station without removing the redundant data) consume more energy of the node in comparison to that of the data aggregation. So to make an efficient use of energy of the nodes, data aggregation is required which is also beneficial in improving the network lifetime [29]. Due to the usage of different data aggregation function like MAX,MIN etc. packets sent to the base station reduced. Thus data aggregation eliminates unnecessary traffic flow of packets within the network and saves the network bandwidth.

1.6 ORGANIZATION OF THE THESIS

The main objective of our research in M.E thesis is to design and develop a centralized energy efficient data aggregation algorithm for WSN. The thesis is organized into 5 chapters including this chapter. Chapter 2 presents a literature review of various data aggregation algorithms. In chapter 3 we have attempted to purpose centralized data aggregation algorithm i.e. Levelling and Self Localization based Clustering (LSLC) with detailed explanation. Chapter 4 covers the simulation results of LSLC protocol as well as its comparison with the existing data aggregation algorithms. Finally, the chapter 5 concludes the thesis and also provide the future prospective of data aggregation algorithm.

CHAPTER 2

LITERATURE REVIEW

In this chapter we have reviewed the literature of different data aggregation algorithms based upon their classification as discussed in chapter 1. The examples of different centralized data aggregation algorithms discussed in this chapter are: Low- Energy Adaptive Clustering Hierarchy (LEACH) algorithm [30], Tiny AGgregation (TAG) approach [8], Power Efficient Data Gathering and Aggregation Protocol (PEDAP) algorithm [6], Energy Efficient Clustering Scheme (EECS) [10], Energy Efficient Clustering Algorithm (EECA) [7], Energy Efficient Sleep Awake Aware (EESAA) algorithm [31], Modified-Low Energy Adaptive Clustering Hierarchy (MODLEACH) algorithm [32], Bandwidth Efficient Cluster based Data Aggregation (BECDA) algorithm [9], Energy Aware Multihop Multipath Hierarchical (EAMMH) algorithm [33], Optimal clustering in circular networks (OCCN) algorithm [11], Cluster-chain mobile agent routing (CCMAR) algorithm [12], Dynamic adaptive hierarchical data aggregation (DAHDA) algorithm [13], Mobile Agent protocol based energy aware data aggregation (MAPA) [14], Scalable and load balanced mobile-agents based data aggregation (SLMADA) [15].

Similarly the examples of different distributed data aggregation algorithm explained in this chapter are: Greedy Approach [16], Exact and Approximate algorithms [19], Ant Colony Algorithm [20], Energy-Efficient and High Accuracy (EEHA) Secure scheme [22], Peony tree based data aggregation (PDA) [25], Real time data AGgregation (RAG) algorithm [23], Clustering and Principal Component Analysis (PCA) based algorithms [21], Privacy Preserving Secure in-network Data Aggregation (PPSDA) algorithm [18], Distributed Data Aggregation Scheduling (DAS) algorithm [17].

The examples of different hybrid data aggregation algorithms discussed in this chapter are: Cooperative data aggregation (CDA) Algorithms[27], Centralized energy allocation (CEA) and Distributed energy allocation (DEA) algorithm [28], Confusion based energy-saving and privacy-preserving data aggregation (CESPT) algorithm [26], Secure data collection scheme based on compressive sensing (SeDC) [24].

Mandeep Kaur, Amit Munjal, *"Data Aggregation Algorithms for Wireless Sensor Sensor Network: A Review"*, Ad Hoc Networks, Elsevier, [SCI, IF 3.047] [Communicated]

2.1 LOW- ENERGY ADAPTIVE CLUSTERING HIERARCHY (LEACH) ALGORITHM

In the literature, there exist a variety of clustering based energy efficient algorithms, one of well known protocol for the clustered environment is given by Wendi Rabiner Heinzelman [30] in 2000 and named as Low- energy adaptive clustering hierarchy (LEACH). LEACH has modified the existing algorithm clustering concept in which once a node become cluster head will remain CH for the entire lifetime of network. Due to this the CH will exhaust its energy very soon. Thereby it reduces the network lifetime. In LEACH, the clustering heads are elected from the all sensor node in every round with certain probability based upon their residual energy.

A node which has higher residual energy among the neighbouring nodes will be selected as cluster head for that round. The cluster head collects the data from all its cluster members, compress it and forward it to base station (B.S).

2.2 GREEDY APPROACH

Chalermek Intanagonwiwat et al. [16] in 2002 explained that in a wireless sensor network, the data dissemination scheme refers to the aggregation of data efficiently at the intermediate nodes with a minimum delay but this is not an energy efficient scheme. So, to overcome the problem of data dissemination scheme, a greedy tree-based approach [16] has been introduced. Greedy Approach is the updated version of direct diffusion approach [34]. In greedy tree-based approach, the method of establishing the path is the same as that of direct diffusion method but there is only one difference that occurs due to the addition of extra energy cost. However, in greedy aggregation approach, two energies are used, one is the energy cost E, that is the energy cost for delivering the event from the source to the current node and another is, incremental energy cost C required for delivering the exploratory event to the existing tree. The proposed approach explains opportunistic aggregation, for the path establishment and repair, as presented in Figure 2.1.

Firstly, the sink node diffuses an interest which is called as an exploratory event. When the nodes match the exploratory event with the data that is present in their cache, then a gradient is being set up. Each node responds to the interest in terms of gradient and is termed as an exploratory gradient. When the source node matches with the interest of the sink then it sends an exploratory event. This event then passes through the multiple paths and reaches the sink. The sink node will select one exploratory gradient from the multiple paths exploratory gradients

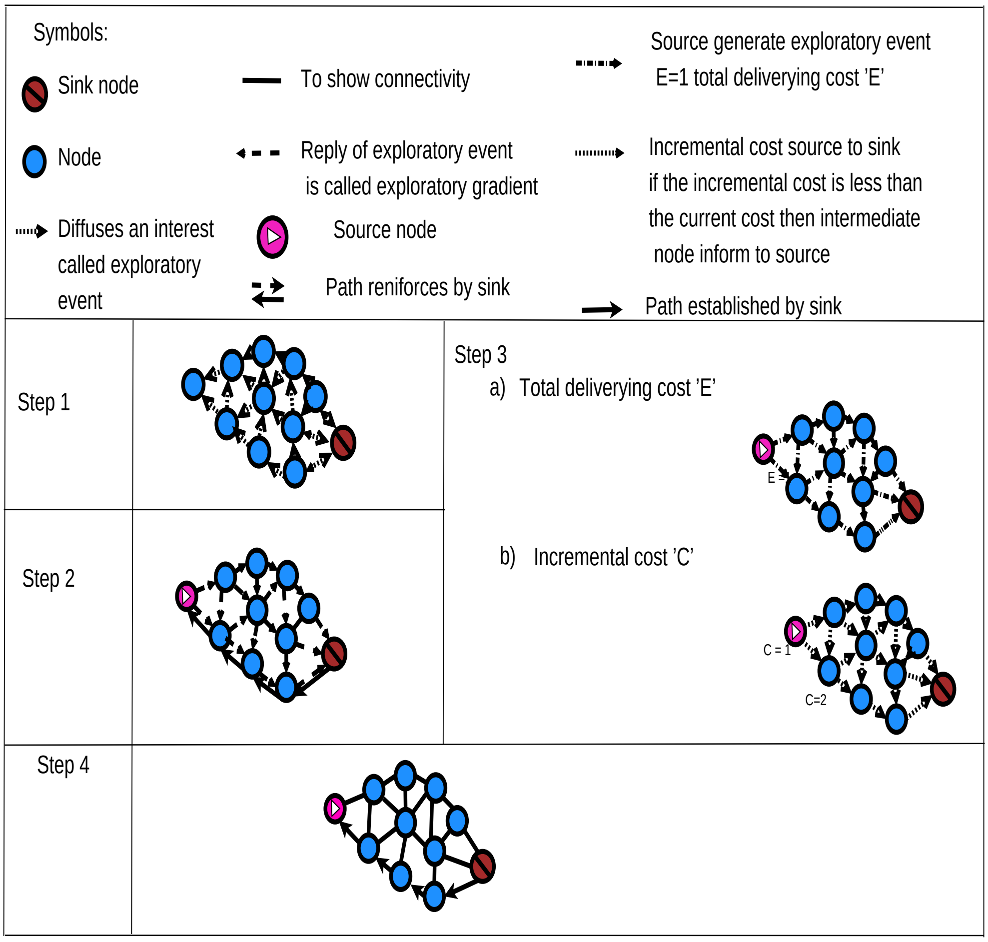


Figure 2.1: Greedy approach

(reply) that provides high quality data.

The basic rule for selecting the path from the multiple paths is on the basis of lower delay (data is reached at sink first) that is used, to reinforce the source. But in case of greedy aggregation, the formation of greedy tree involves energy cost E for delivering the event from one node to another. When the source node detects the exact data as per the interest of the sink then it will generate an exploratory event with E=1 towards the nodes for which it has a gradient. Once, the next node receives an exploratory event and it does not have the information regarding the interest of sink then, the node will act as an intermediate node. The intermediate node will add the transmission cost to E if it happens to be in the path between the source node to sink node.

Thus, the value of E will keep on increasing till the sink node is reached. Thus, with the help of E, we can estimate the total delivering cost from the source to sink. But, this cost is not sufficient enough to construct a greedy tree as path sharing information is not involved.

Hence, to provide this information, the incremental cost message is also included. This message includes two fields: random message id & energy cost C required to connect the exploratory event to the existing tree. When the intermediate node receives the incremental cost message, then it will check its cache for the exploratory event contained in the message. In case, it finds a new source whose incremental cost is less than the existing source then it will forward the same to the sink node. The sink node will then wait for the expiry of timer t_p after which it will reinforce the path.

In the simulation, the author performs the comparative evaluation of greedy aggregation approach with opportunistic aggregation for data dissemination on the basis of average dissipated energy per packet as a function of the network density. The impact of network dynamics is also performed. The results show that for low density networks the greedy aggregation and opportunistic aggregation are almost equivalent to each other in terms of energy savings. But considering the high density network, proposed approach shows efficient energy compared to opportunistic aggregation approach without any harmful effect on delay.

2.3 TINY AGGREGATION (TAG) APPROACH

In [8] Samuel Madden et.al in 2002 have proposed a distributed low power data aggregation scheme which is termed as TAG i.e Tiny Aggregation. The TAG is a query based scheme which is used in adhoc networks. The TAG is mainly composed of two phases, distribution phase, and a collection phase. In the first phase, the queries are distributed among the network. In the second phase, gathering of data using structured query language (SQL) from the sensors. The SQL helps to identify the key features of aggregation functions.It has been analyzed that TAG helps to reduce the communication cost and it also has an ability to tolerate the disconnections. The TAG was evaluated on the basis three models: simple, random and realistic model.

The performance of TAG is compared with the centralized approach based upon different aggregate functions such as *MAX, COUNT, AVERAGE, MEDIAN*etc. It was analyzed that TAG performs better than the centralized approach and can help to reduce the communication cost. The Authors have also presented many techniques for the improvement in the performance of TAG. Those techniques are taking an advantage of a shared channel, hypothesis testing etc. and hence improve the tolerance to the loss of TAG. This scheme was implemented in real using Tiny OS mica motes and it was observed that TAG performed better over the centralized approach.

2.4 POWER EFFICIENT DATA GATHERING AND AGGREGATION PROTOCOL (PEDAP) ALGORITHM

In [6], Huseyin Ozgur Tan and Ibrahim Korpeoglu in 2003 proposed two data aggregation algorithms, PEDAP (power efficient data gathering and aggregation protocol) and PEDAP-PA (power efficient data gathering and aggregation protocol-power aware). In both these algorithms, the minimum spanning tree is used as a routing scheme. In the first algorithm, authors have used a minimum spanning tree for computing the route from source node to base station node. The route cost is computed based on two types of cost: one is node to node cost that is involved in the route and another one is node to base station cost.

In PEDAP-PA, minimum spanning tree which is formed to create a root from the source to the base station depending upon the cost between node to node and node to base station. Residual energy of the node is also considered to make the minimum spanning tree. Also, node with higher residual energy will act as data aggregator node. After a certain time period when the residual energy of the node below a certain amount reformation of the minimum spanning tree is performed, node with less residual energy will not take part in the minimum spanning tree formation and will act as a leaf node. This algorithm is the power aware version of PEDAP, as both the algorithms are shown in Figure 2.2

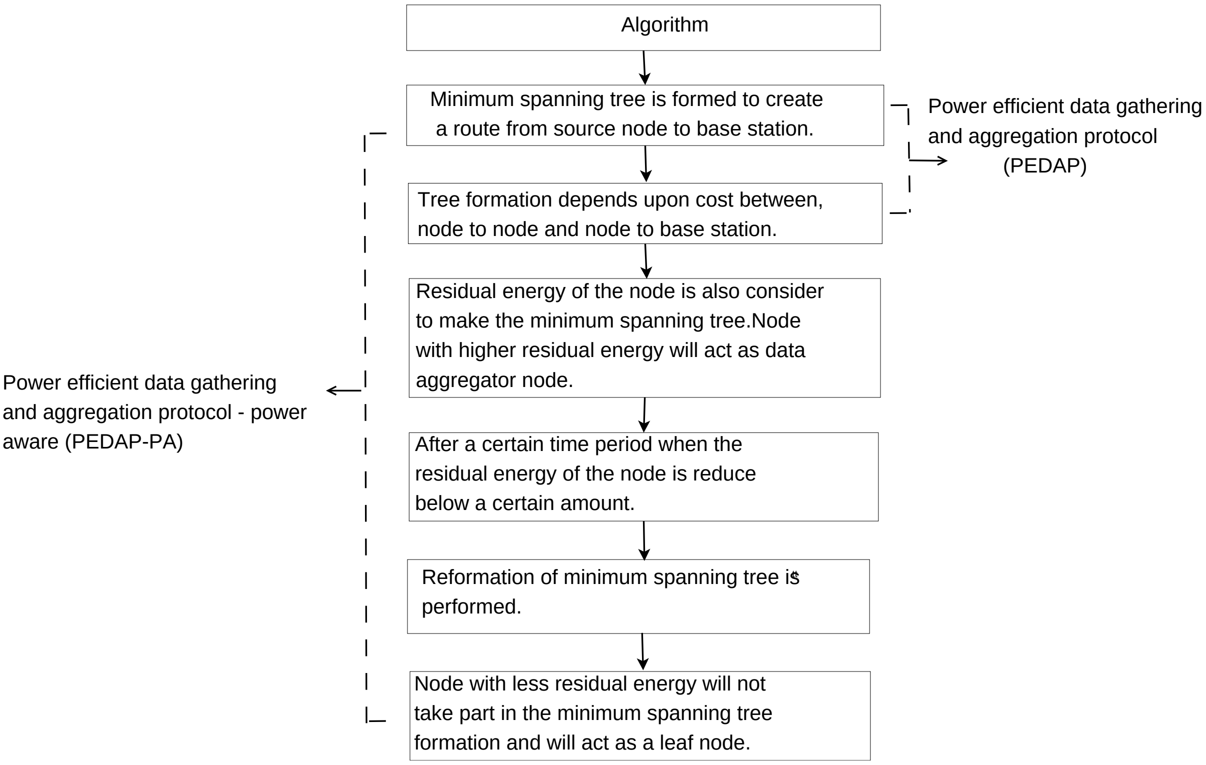


Figure 2.2: PEDAP and PEDAP-PA algorithms

The simulation results show that if the load is evenly distributed among the nodes then the lifetime of the network is increased. Both the algorithms provide good results when the base station is inside the field. When all the nodes have to work together than PEDAP-PA is performed best than LEACH and PEGASIS. On the other hand, if the lifetime of the last node is important than PEDAP is a good approach. Thus PEDAP and PEDAP-PA performs better than the existing protocols such as LEACH [30], PEGASIS (power efficient gathering in sensor information system) [35] in terms of lifetime enhancement.

2.5 EXACT AND APPROXIMATE ALGORITHMS

In [19] Jamal N Al-Karaki et.al in 2004 have discussed that the data aggregation is the basic approach in the WSN to enhance the lifetime of the network and it is affected by many factors such as:

- Selection of the node at which data is to be aggregated.
- Data aggregation function.
- Impact of the network density on data aggregation.

These algorithms tried to solve three problems faced in data aggregation,

1. Lifetime maximization by selecting optimal aggregation point.
2. Routing and data aggregation as a joint problem.
3. Reduced clustering and routing overheads.

They have also discussed the tradeoff between energy savings and the delay included during the time of data aggregation. Basically, they have used the hierarchical model as shown in Figure 2.3 to perform the data aggregation. In this model, two level of aggregation is performed firstly, at the level of some selected nodes called local aggregators (LAs) and then at the second level, aggregation is performed on the level of master aggregators (MAs), where MAs are a subset of LAs.

During the simulation, they have analyzed the three points which are discussed as below,

1. Firstly, the k means [36], genetic algorithms [37] were analyzed for aggregation data v/s non aggregation data. It was observed that k means approach provides best results and greedy approach provides good results.
2. The number of MAs increased and so the enhancement of network lifetime is achieved.

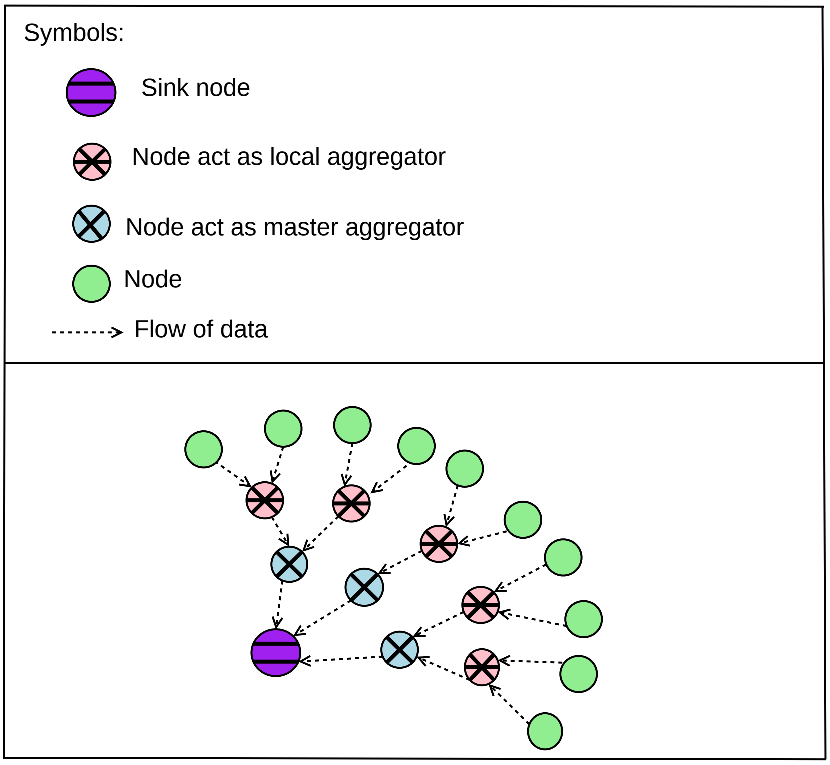


Figure 2.3: Exact and Approximate algorithm

But this is true up to a certain level after which the network lifetime becomes invariant with an increase in M.A.

3. When the data is transferred from multiple LAs to MA, the difference in the minimum and the maximum number of hops traversed is termed as aggregation delay. The two parameters which affect the delay are node transmission range, the number of sensor nodes in the field. It is analyzed that aggregation delay is inversely proportional to both the transmission radius as well as the number of sensor nodes. It was concluded in the result that just to select the master aggregator they have introduced the exact and approximate algorithms so that network lifetime can be increased. Approximation approach produces the results which are almost equivalent to optimal solutions.

2.6 ENERGY EFFICIENT CLUSTERING SCHEME (EECS)

Mao Ye et al. in 2005 proposed an energy efficient clustering scheme (EECS) [10] for enhancing the lifetime of WSN. The basic approach in this algorithm is to decrease the energy consumption of the nodes at the time of data aggregation using cluster based approach. One of the famous examples of energy saving is LEACH [30]. The application where periodic data aggregation needs to be performed EECS is more suitable than LEACH.

The basic difference between EECS and LEACH is that EECS comes under action at the time of selection of cluster head and it also has the concept of load balancing between the clusters. In the EECS, initially Base Station (BS) broadcasts the message to all the nodes. With the help of received signal strength (RSS), all the nodes will compute their distance from the base station. Each node after computing their distance will set up their radio range.

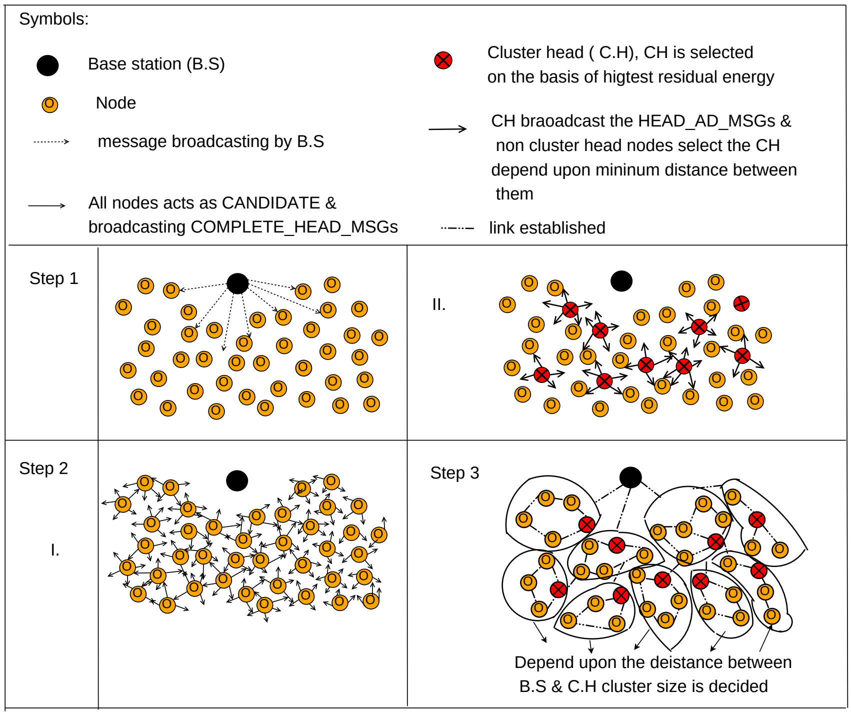


Figure 2.4: Flow diagram of EECS algorithm

In EECS cluster based scheme, two steps are involved: Cluster head election and Cluster formation. In the cluster head (CH) election, every node acts as a *CANDIDATE* and broadcasts the *COMPLETE_HEAD_MSGs* as shown in Figure 2.4, within its radio range and before a certain threshold value of time. Every *CANDIDATE* node will check the message of every other *CANDIDATE*, If the candidate has less residual energy then it leaves the competition without receiving any further *COMPLETE_HEAD_MSGs*. In the second step, cluster formation is performed in which every *HEAD* node broadcasts the *HEAD_AD_MSGs* in the network. Then the non-CH nodes will decide their cluster head. Mostly non-CH selects the cluster head based upon minimum distance towards the cluster head in order to save their energy. The cluster head node's responsibility is to aggregate the data of all its non-CH nodes and thus the energy consumption at cluster head increases. The energy consumption at the cluster head is a combination of three elements:

1. Data Reception

2. Data Aggregation
3. Data Transmission

The major contribution of energy consumption of CH node is for data transmission to the base station. As it involves the distance between base station and cluster head node. In this algorithm, the authors have proposed a variable size of different clusters. The size of the cluster is dependent on the distance of it's CH to the base station. If the distance between the base station and cluster head is larger, than the size of the cluster (i.e. less number of nodes in that cluster) should be reduced. The authors have also proposed a load balancing by which every non-CH will associate with a CH on the basis of the weighted cost function. This involves two parameters: one is a distance of non-CH to CH and another is a distance between CH to the base station.

The results are obtained for two different scenes, the normal scene in which the number of nodes is taken to be 400 to 800 and the other is a larger scene in which the number of nodes is taken to be 1000 to 1500. In the simulation, the authors measured the lifetime in terms of round till the first node dies for the proposed protocols as well as for the LEACH.

Firstly, the impact of T (threshold between 0 and 1) on the network lifetime is analyzed with the normal and large scene. Later on, the network lifetime is analyzed by varying the radio range of the candidate node. And finally the results are computed with and without cost function for normal scene and large scene respectively. All the above evaluations are performed for EECS and LEACH [30] protocol. It is analyzed that EECS provides 35% better lifetime than LEACH and energy usage in EECS is 93% and in LEACH is 53%. This is because the cluster heads are chosen on basis of residual energy. Thus, EECS increases the lifetime of a network in comparison to LEACH [30].

2.7 ANT COLONY ALGORITHM

Wen-Hwa Liao et al. [20] in 2008 proposed an ant colony algorithm, as shown in Figure 2.5. They have worked upon increasing the probability of searching the aggregation nodes using two routing paths for ant colony algorithm [38].

The algorithm can be explained using three steps. The first step is about the selection of the next hop node. Flooding process is performed by the sink node towards the all other nodes present in the network. When all the nodes receive this packet, the hop count to the sink node is calculated. When the source node is about to forward the data to the sink node, then

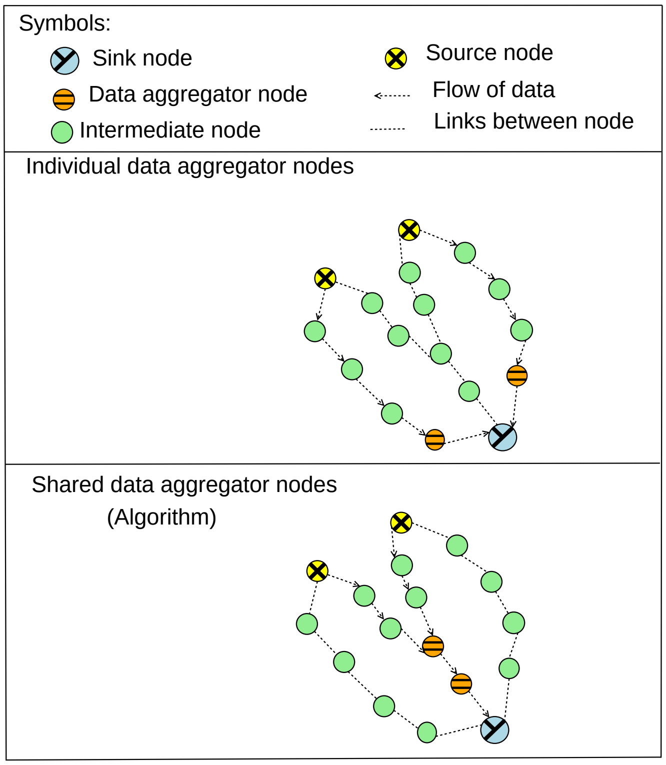


Figure 2.5: Ant colony algorithm

the next hop node for the source node is calculated using the random proportional rule [20]. In the second step, mechanism of increasing the probability of finding the aggregation nodes is discussed. The source node forwards the packet to the next selected hop. The format of the sent packet contains six different fields including selected node (SN), source node id (S), previous hop node id (P), hop count from the node i to the source nodes (D_{is}), extended hop count (EHC), time to live (TTL). D_{is} is zero initially. When the node i forwards a packet to node j it will generate the reaction table. The reaction table contains five fields. All those nodes which already have the reaction table will only check the source id (S) of the selected packet with the stored source id in the table. After matching the value of S, if values comes out to be the same then node j will select the lower value of D_{js} and update its reaction table, Otherwise, the new record entry is made in the reaction table corresponding to the selected packet. Once the node j has been updated in the reaction table.

Then, it will broadcast the information about the data selected packet but not about the data packets to its neighbours. The Time to live (TTL) of the selected packet is computed by decrementing the TTL of the reaction table by one. When TTL becomes equal to zero, at that time the node j will stop finding the path. In the third step, authors have performed pheromone update. This is done by comparing EHC with that of TTL. In case the EHC matches with TTL then the node will transmit pheromone_updated packet. This packet contains the information like the node id (ID), id of the previous hop node (P), the hop count of the node j to the sink node (h_j) and the total cost of the source reaching to the sink node through node j. When the node i receives a pheromone_update packet from the node j, it will update its pheromone table. This is done with the help of computing the difference of the hop count from node i to sink node h_i as well as node j to sink node h_j. If $h_i > h_j$ then we deposit pheromone from i to j, if $h_i = h_j$ then after which it will depend upon $\Delta\omega_j$ pheromone. Otherwise, if $h_i < h_j$, then there is no need to deposit the pheromone. At the aggregator node, the pheromone levels of all the neighbors are being updated. In case no ant visit this node, then its pheromone get evaporated. Otherwise, the pheromone level at aggregation node increases in such a way so as to attract ants carrying different data from the other sources.

The authors have simulated their protocols with that of direct diffusion (DD) [39] method based upon three different performance metrics that include total energy consumption, average energy consumption and maximum lifetime of the network. It means that energy consumption of Ant-1, Ant-2 is very less as compared to the Ant-0 and DD because there more aggregation nodes are present to deliver the data. As the number of source node increase, the average energy consumption gets decreased because more the number of aggregation nodes in the network, the lesser is the energy consumption. Results are obtained by comparing the DD method with proposed ant colony algorithm by varying the source node. It was found that Ant-1 and Ant-2 are better than Ant-0 and DD because more the number of aggregation node, the higher will be the energy saving. Thus it is concluded that very less energy is consumed in the algorithm for data delivery as compared to that of DD method.

2.8 ENERGY EFFICIENT CLUSTERING ALGORITHM (EECA)

The Cluster based approach is generally used for data aggregation as it is energy efficient. Earlier the clustering algorithms selects the cluster head without considering the whole network. This clustering may not be energy efficient so SHA Chao et al. [7] in 2010 discussed an energy efficient clustering algorithm (EECA). In EECA, all the nodes broadcast the message within their communication radius and the broadcasted message contains different parameters such as node id, residual energy and it's coordinates. All the nodes which receive this message will update their neighbour information table. The attributes of the neighbour information table are node id, neighbour node coordinate and residual energy. The node can participate in cluster head election if it's residual energy is greater than the threshold. The cluster head candidate will compute the mean distance of all it's neighbour nodes and it will broadcast the cluster head competition message which consist of the competition bids. The competition bids contains the

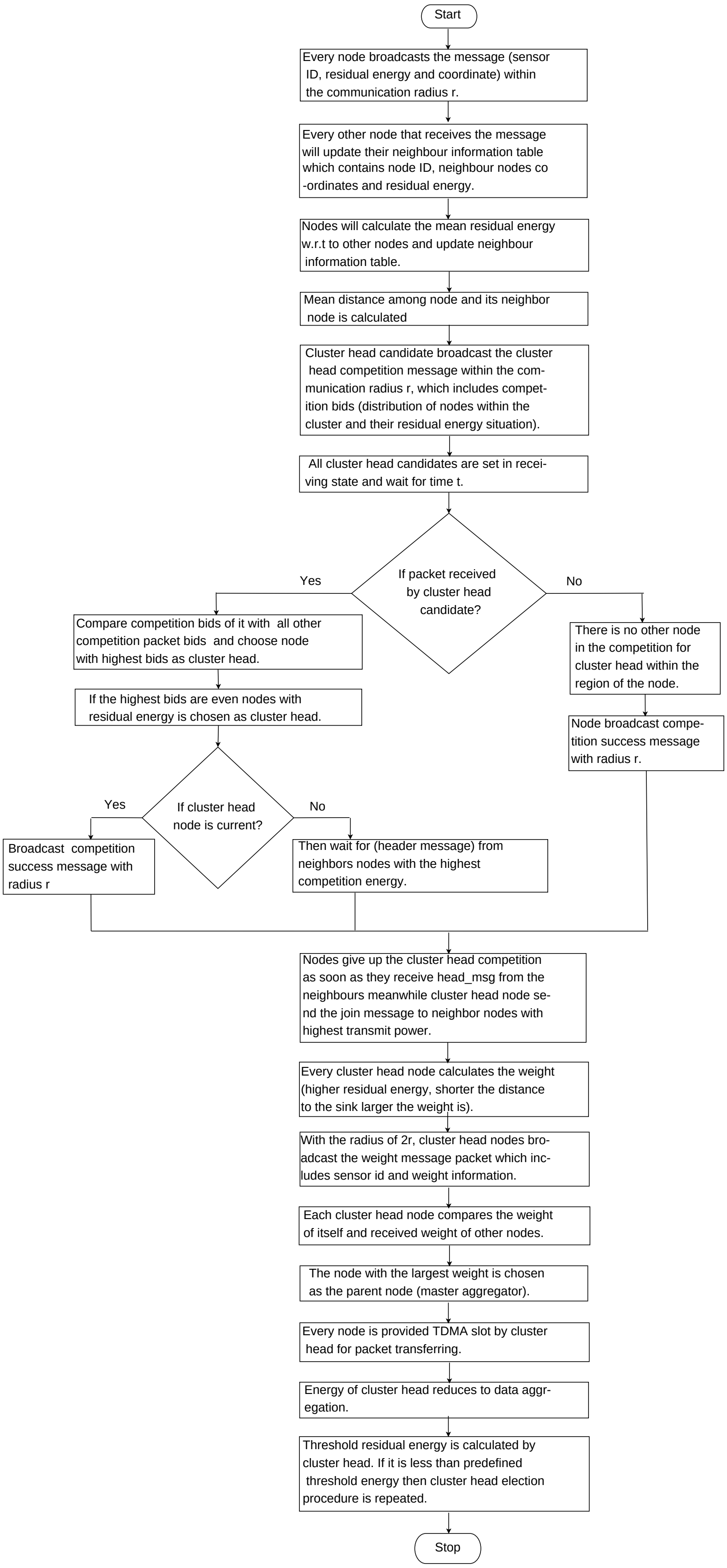


Figure 2.6: Flow chart of EECA algorithm

distribution of the node within the cluster and their residual energy.

All the cluster head candidates will wait for the expiry of time T. In case a packet is received by any of the cluster head candidate after the expiry of time T, then competition bids of that cluster head candidate are compared with all other competition bids. The cluster head candidate which has the highest bids is chosen as cluster head. If the highest bids are even than node with residual energy is chosen as a cluster head. The selected cluster head will broadcast *Head_Msg* within in it's communication radius and all the other nodes that receive *Head_Msg* will give up the cluster head competition. Meanwhile, the cluster head node sends the join message to the neighbour node with highest transmit power.

Moreover, every cluster head calculates its weight depending upon its residual energy and the distance from the sink. In addition to this the cluster head node broadcasts the weight message packet which includes sensor id and weight information. Thereafter every cluster head compares it's own weight with that of received weight. The node with the largest weight is chosen as parent node of the compared cluster head node. In this manner, a data aggregation tree is formed. Every node is provided with the TDMA slot by its respective cluster head for transferring the packets. The energy of cluster head reduces due to the data aggregation. The Cluster head calculate it's threshold residual energy in each round and if its energy becomes less than the predefined threshold energy then the cluster head election procedure is repeated as shown in the flowchart in Figure 2.6.

The performance of EECA is analzed by comparing it with LEACH [30] algorithm. The Comparison is performed on the basis of three parameters: network lifetime, average residual energy, variance of residual energy. On the basis of network lifetime, it is analyzed that EECA algorithm performs better than the LEACH because in EECA clustering algorithm, the distance among the cluster heads is reduced which leads to the reduction of energy. On the other hand, in LEACH, clusters head are randomly chosen with probability without keeping in mind the residual energy of the nodes. Thus, the network lifetime is increased in EECA. The results have proved that the average residual energy of nodes in EECA is more than that of LEACH because of the proper clustering head mechanism and balancing of the energy of nodes, in EECA. EECA algorithm also performs better in term of variance of residual energy than LEACH.

2.9 COOPERATIVE DATA AGGREGATION (CDA) ALGORITHMS

Hongli Xu et al. [27] in 2010 has discussed the cooperative data aggregation (CDA) problem and proved that it is NP-hard. They also provide two cooperative data aggregation algorithms:

centralized maximum cover tree (CMCT or MCT), distributed maximum cover tree (DMCT). CMCT algorithm consists of two steps. Firstly, minimum spanning tree is formed using Kruskal's method [27]. Every node also determines its parent node. Secondly, every link between the node and its parent node in the minimum spanning tree is marked as unvisited. Then the cooperative pair selection is performed on the minimum spanning tree. The cooperative pair selection is the method by which the different nodes will coordinate themselves for transmitting their packets. The weights for all the unvisited links are computed if the calculated weight of the link is less than or equal to zero then that link is marked as visited otherwise the link remains unvisited. In the next iteration, the MCT algorithm selects an unvisited link having maximal weight and is called visited link and corresponding all the other unvisited links attached to the visited link are also marked as visited. The algorithm performs multiple iterations till there is no unvisited link left in the MCT. The DMCT algorithm explained by authors comprises of mainly three steps. In the first step, the distributed minimum spanning tree is constructed with the help of distrusted Gallagher-Humblet-Spira (GHS) [27] algorithm.

In the second step, weight adjustment is performed with an assumption that every node knows its parent node. For every leaf node, the weight is calculated and that weight is sent to the leaf node after some delay by the parent node. When the weight of all the links is calculated then the parent node merges all these weights to one packet and sends that packet to its one-hop neighbour. In the third step, cooperative pair selection is performed in which all the links are marked as unvisited. After that, an unvisited links having the largest weight is selected. If the selected link is an out edge of the parent node then that link is marked as visited and a *WIN* message is sent to its one-hop neighbour and the algorithms finished. All the nodes that receive the *WIN* message and are linked to either side of the visited link are marked as visited. when all the links are marked as visited then the algorithm terminates. As both the MCT and DMCT algorithm forms the same network topology so the authors have compared the results for MCT algorithm only. The performance of MCT algorithm is compared with PEDAP [6] and PEGASIS [35]. It has been analyzed that in MCT algorithm the power consumption is decreased as compared to PEDAP and PEGASIS. It is also analyzed that as the number of sensor nodes increase as the power consumption decreases. Hence, MCT algorithm performs better than PEDAP and PEGASIS.

The clustering environment plays an important role in performing data aggregation for any wireless network. Each cluster contain cluster head (CH) which helps to aggregate the data collected from other sensor nodes. The data aggregation is further helpful in reducing the redundant data from the sensed data by the nodes. This allows only the required data to be transferred to the sink and thus lead to reduce energy consumption of nodes and thus network lifetime is increased.

2.10 ENERGY-EFFICIENT AND HIGH ACCURACY (EEHA) SECURE SCHEME

The EEHA secure scheme [22] has been introduced by Hongjuan Li et al. in 2011 in which the data aggregation can be performed in such a way that the energy consumption is minimum and the data accuracy is higher. EEHA is described in four steps and described in Figure 2.7

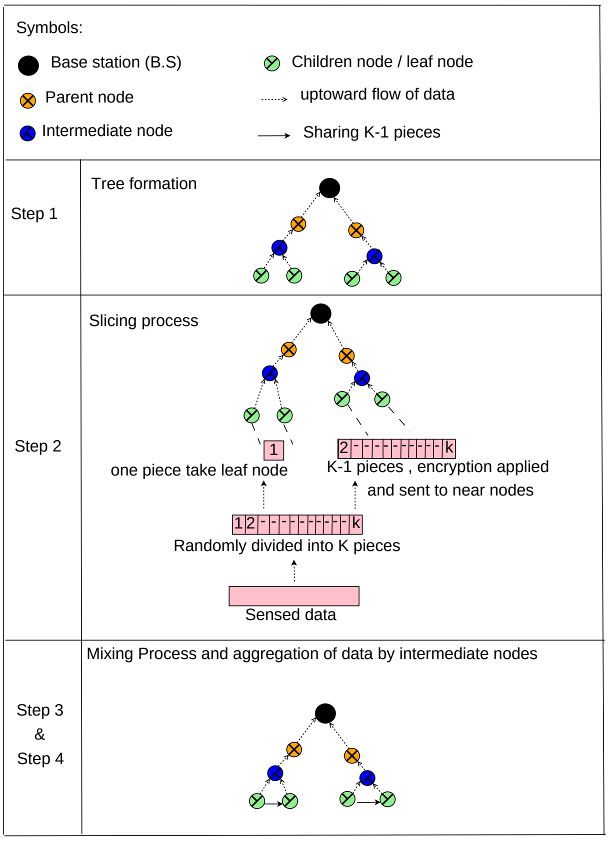


Figure 2.7: EEHA algorithm steps

In the first step, aggregation tree is constructed which is directed towards the base station. This is done by taking the union of different paths from each sensor node to the base station. The data aggregation tree so formed may not be the shortest path tree. In the second step, Slicing is performed on the aggregation tree. In which each leaf node randomly selects a set of the nodes within h the hop distance. The Leaf node senses the data and randomly divides the data into k pieces. In the slicing process, leaf node takes one piece from the k pieces and other k-1 pieces are sent to the other nodes. On these k-1 pieces, encryption is applied and the encryption key is only known to the sender and the receiver node. Aggregation results are obtained by the summation of the sensor nodes data. In the third step, the mixing process is performed. In this step, all the leaf nodes will wait for a certain amount of time in order to ensure that slices reach at the receiver node. Then all leaf nodes will decrypt the data using the key and then add k-1 pieces of its own to it. In the fourth step, the node will sum up all the slices then encrypt them and finally send to their respective parent nodes. Intermediate nodes perform the aggregation on the data that it receives on the leaf node. It means parent nodes have to wait for a long time for the data. Then intermediate nodes send the data to the parent node within a specific time. Finally, the whole data that is collected from all other nodes is summed up at the base station.

The performance of EEHA algorithm is compared with another protocol known as Slice-Mix-AggRegaTe (SMART) [40]. The evaluation parameters considered are communication overhead, aggregation accuracy, the efficiency of privacy. In case of communication overhead, EEHA algorithm performs better than the SMART. This is because the bandwidth consumption in EEHA is less due to a reason that exchange of packet is the network in a smarter way. This leads to a decrease in the communication overhead in EEHA. Aggregation accuracy of EEHA algorithm is better than the SMART as there is less communication overhead in EEHA due to which collision between the messages is less. Privacy preservation of EEHA algorithm is moderate but still better than the SMART because security is provided at the leaf node by slicing the messages rather than the intermediate node provide security by aggregating the messages. Obviously, energy consumption of EEHA is less because overhead is less than the SMART. Overall it is concluded that EEHA algorithm provides better results than SMART.

2.11 PEONY TREE BASED DATA AGGREGATION (PDA)

Pei Wang et al. [25] in 2011 introduced a PDA scheduling algorithm which emphazises on improving the latency of the data aggregation in WSN. The PDA scheduling algorithm consists of three stages: Initially, the pony tree is constructed so that it covers all the nodes and form a maximum independent set. The nodes which come under the maximum independent set is called as dominative nodes. To connect the dominative nodes with each other some extra nodes are selected is termed as connective nodes. Some of the nodes which do not come under dominative node and connective node are called as white node. In the introduced algorithm, to avoid the collision as well as latency, the minimum number of connective nodes are selected.

In the second phase, local data aggregation is performed in which white nodes sent the data to the dominated node at different time slots so that collision of data packets, as well as the latency, is reduced. The dominative node performs the local aggregation of data. In the third stage, dominative nodes sent data to the connective nodes where aggregation is performed again and is called global aggregation. The performance evaluation of PDA is performed by considering identical communication range and identical interference range for all the sensor nodes. The comparison of PDA is carried out with the algorithm of Huang et al. [41]. Which shows that aggregation latency is reduced by 20% in comparison to the existing algorithm. It means PDA provides considerable results in term of latency.

2.12 REAL TIME DATA AGGREGATION (RAG) ALGORITHM

Hamed Yousefi [23] et.al. in 2012 explained that for data gathering applications basic structured approaches are used. But due to the use of these approaches in dynamic scenarios, the maintenance overhead increases. Also, in case of critical applications, WSN would have to complete the derived task in a particular interval of time and will have to collects the useful information. This is the problem faced by WSN when they are used in real time application. To overcome this problem, authors have proposed a structure-free RAG protocol. The RAG is formed with the help of two policies a) Judiciously waiting policy b) Real time data-aware anycasting policy, as shown in Figure 2.8. In Judiciously waiting policy, it is explained that to aggregate the data at the intermediate node, some packets that reach early have to wait for the other packets. Hence, the main issue in real time monitoring is for how much time intermediate nodes have to wait for the other packets so that aggregation gain can be maximize. There is a particular time limit during which the intermediate nodes can send data to the sink. So how to maximized the on-time end to end delivery ratio is considered as of the issue.

The solution to the above discussed problems is provided as Judiciously waiting policy. According to this policy, artificial delays (slack time) are being introduced in those packets which are transmitted toward the sink and in turn perform effective aggregation due to which timing process is increased. In real time applications, there is a time for serving the packet which is termed as TTD (time to deadline). And it plays a vital role in making routing decisions. The slack time in the available packets i.e. TTD-EED (time to deadline-end to end delay) is allocated proportionally based upon remaining hop count towards the sink node. The EED is the estimated end to end delay which is defined as a time that is used to deliver the packet from the current node to the sink node. In order to estimate the values of the end to end delay (EED), they have estimated one-hop delay and thereby compute the waiting timeout (it is the time required by each forwarding packet to reach at the sink within a certain time limit). As the packets move close to the sink then the hop count will be decreased and slack time is used for performing the data aggregation. At the end, the packet spends all its remaining deadline time as a slack time if the next node is a sink node.

Thus, judiciously waiting policy is able to improve the aggregation efficiently and also one can calculate the waiting timeout which is very important for the real time applications. In real time data-aware, Anycasting policy selection of the next hop plays an important role in real time dynamic routing. So aggregation of data in real time is a prime requirement that needs to be fulfilled. To make real time decision, the current node needs to calculate the velocity before it sends the packet to the sink so that TTD of the packet can also be achieved. If the required velocity of the packet for each hop is maintained then the TTD is achievable. Thus, this policy is able to consider the current network situation and also adapts to the changes. If the sink node is away from the source, the velocity of the packet will be increased and vice versa.

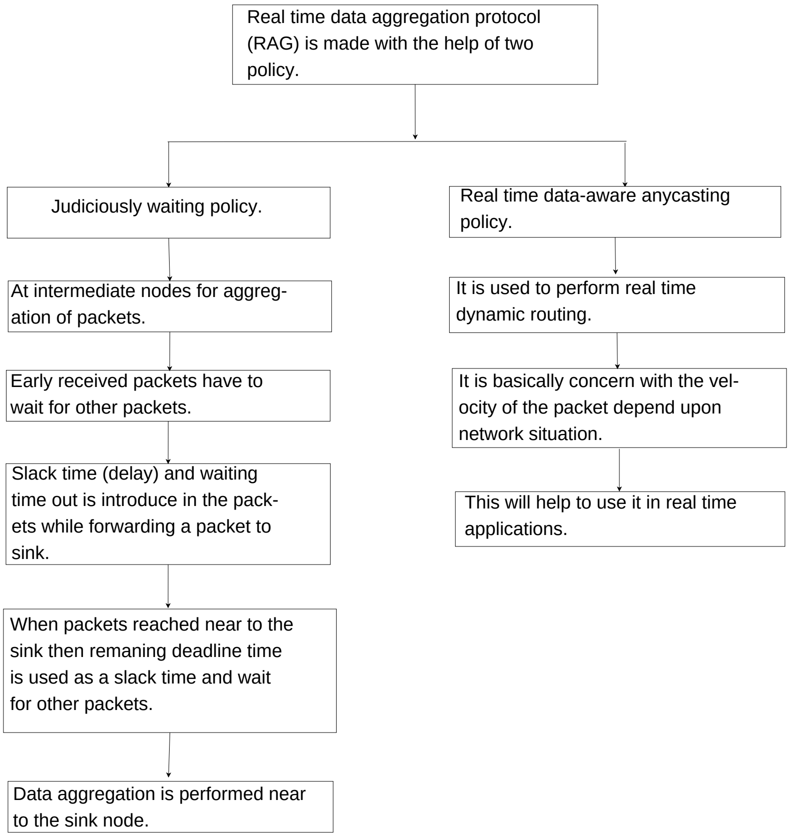


Figure 2.8: Flow diagram for RAG algorithm

Performance evaluation of the real time data aggregation (RAG) is performed on the basis of aggregation gain, miss ratio, energy consumption and the end to end delay of WSNs. The RAG and Stateless Protocol for Real-Time Communication i.e SPEED [42] are being compared on the basis of aggregation gain, miss ratio, energy consumption and the end to end delay of WSNs. The simulation results show that RAG performs better over SPEED at the higher data rate. In RAG, as the data rate is increased, energy consumption decreases but the miss ratio is increased, this is due to heavy traffic. So in order to overcome the incremental change in the miss ratio, the authors have increased the aggregation limit i.e more number of packets are now aggregated to form one packet.

Thus, it will lead to a decrease in the miss ratio. Hence, RAG performs better than SPEED in terms of energy consumption and miss ratio for different data rates. End to end delay between SPEED and RAG protocol is also analyzed with different traffic loads and the result proved that RAG performed better than the SPEED have but RAG has the higher end to end delay. It means RAG protocol delays the packet at every node so that proper aggregation of data is performed.

2.13 ENERGY EFFICIENT SLEEP AWAKE AWARE (EESAA) ALGORITHM

T.Shah et.al [31] in 2012 explained an energy efficient sleep awake aware (EESAA) protocol. The main aim of this protocol to reduce the energy consumption by using the paring concept. The paring concept allows the nodes of similar application to form a pair (if they are near by) for sensing the data as well as to communicate it to the base station. In the network, nodes with the same application will make their pairs with their near by nodes. The data sensing from the environment can be performed by paired nodes using the concept of *Sleep/ Awake*. The paired nodes are initially in *Awake* mode. The paired nodes will compute their distance from the CH as well as between them. CH selection procedure is similar to LEACH.

The node whose distance is less from the CH will sense the information from the environment and will send it to the base station. While the other nodes will remain in *Sleep* mode. During next round the nodes will exchange their responsibilities i.e. the earlier node will go to *Sleep* mode and the other node will become Active. Thus energy consumption of the network is also minimizes and network lifetime is enhanced. Simulation results shows that EESAA performs better than A stable election protocol (SEP) [43], LEACH [30], Distributed energy efficient clustering scheme (DEEC) [44].

2.14 MODIFIED-LOW ENERGY ADAPTIVE CLUSTERING HIERARCHY (MODLEACH) ALGORITHM

In [32] D.Mahmood et.al. in 2013 discussed the modified version of basic LEACH protocol and is named as MODLEACH. The basic concept of MODLEACH is similar to that of LEACH but the only difference is during cluster head reelection process. Earlier in basic LEACH the cluster head is reelected in every round where as in MODLEACH the author have introduced the concept that if the existing cluster head has sufficient residual energy i.e. more than required threshold then there is no need for cluster head reelection. This means the existing cluster head will continuous to remain as cluster head for the subsequent round as long as its residual energy is greater than the threshold energy. This concept reduces the packet overhead required for the cluster head reelection procedure and thus the energy consumption of the nodes is minimized.

They have also introduced the concept of amplification in which they have discussed that when a node becomes cluster head it will increase its transmission power level to deliver its packets to the base station or to near by cluster head so that it will transmit the packets to the base station. Now in the next round, if the cluster head becomes normal node then it will decrease its amplification power level. Finally, the authors have also introduced the concept of soft threshold [45] and hard threshold [45], that decides whether to transmit the sensed attribute to base station or not. It is analyzed that MODLEACH performances better than LEACH [30] in terms of alive nodes, dead nodes of network, number of cluster heads, packets transmitted to base station.

2.15 CENTRALIZED ENERGY ALLOCATION (CEA) AND DISTRIBUTED ENERGY ALLOCATION (DEA) ALGORITHM

Shilang Xiao et al. [28] in 2014 introduced two algorithms that are centralized energy collection (CEA) algorithm and Distributed energy allocation (DEA) algorithm. The CEA algorithm is based upon immune-genetic heuristic [46]. The aim of this CEA algorithm is to find the optimal energy scheme so that aggregated data received at sink should be accurate. The CEA algorithm is executed by the sink nodes. The results proved that CEA algorithm provides accurate results for smaller size network but for larger size network the results are not accurate. This is because in larger size network frequent node failure occurs due to which the topology changes occur very rapidly. This increases the communication overhead. So to make the algorithm feasible for larger size WSN, authors introduced DEA algorithm in which Gibbs sampler [47] theory is used.

The performance analysis shows that in larger size network, DEA performed better in terms of accuracy of the aggregated data received at the sink. Further, the comparison of CEA and DEA is performed with algorithm including greedy transmission (GT) [46] and OPD (Optimized power declaration) [48] algorithm. The results are obtained by varying the link loss probabilities and it is analyzed that CEA and DEA is superior than algorithms in term of data aggregation precision (accuracy) i.e. they are more accurate for larger size network.

2.16 BANDWIDTH EFFICIENT CLUSTER BASED DATA AGGREGATION (BECDA) ALGORITHM

The BECDA has been proposed by Dnyaneshwar S. Mantri et al. in 2015 for an efficient utilization of available bandwidth [9]. The authors proposed the algorithm in which data gathering is performed in such a way that bandwidth is efficiently utilized. There are few assumptions considered for the node as well as the network. The network is assumed to be heterogeneous in nature with uniform density and random distribution of static nodes with the sink node assumed to be mobile. There exist three kind of nodes i.e normal, advanced and super nodes. Another assumption for node point of view is that the count of nodes for all the three types is same. Moreover, all the nodes are generating data at the different rate.The data is generated by every node in the range of 0 to 1. Based on all these assumptions the algorithm is formed which consists of three phases.

In the first phase, clusters are formed from the randomly placed heterogeneous nodes. After that, from each cluster, a cluster head (CH) is elected for an inter cluster and intra cluster aggregation. The selection of the CH depends on the basis of highest energy among the cluster members as well as highest average one-hop neighbours. In the second phase, each CH will perform intra cluster aggregation by aggregating the data of variable rates while keeping the fixed packet size from all its cluster member. In the third phase, each CH acts as a normal node to perform inter cluster aggregation. The CH with highest residual energy will forward the aggregated data towards the sink nodes as shown in the flowchart of Figure 2.9.

The aggregation in this algorithm is performed twice so the number of packets sent to the sink get reduced. The analysis of BECDA algorithm is performed for two different conditions i.e. either each node in the network will generate an equal number of packets or each node in the network will generate a different number of packets. The performance of BECDA algorithm is compared with two different algorithms: Two-tier cluster based data aggregation (TTCDA) [49] and Energy efficient clustering and data aggregation (EECDA) [50] algorithm. The results show that the throughput (no.of bits received by the sink) for BECDA is small in comparison to that of TTCDA and EECDA for both the conditions.

This is because in BECDA redundant, the packets are removed at the aggregation stage of inter cluster as well as at intra cluster aggregation stage. The average energy consumption ratio i.e.

$$\frac{sum\ of\ total\ energy\ comsumption}{no.\ of\ nodes}$$

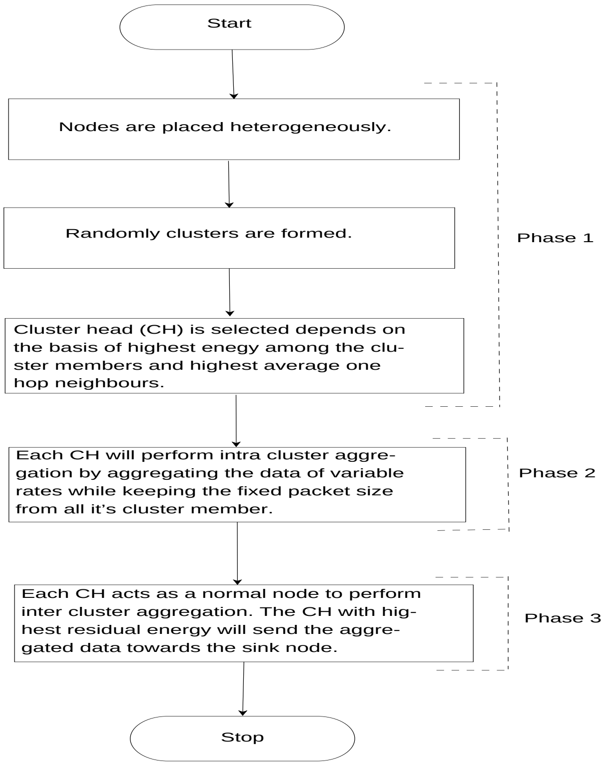


Figure 2.9: Flow diagram of BECDA algorithm

is less in both the conditions in comparison to EECDA but it is more in comparison with TTCDA. This is due to the reason that aggregation is performed at two stages, inter cluster, intra cluster stages and hence less number of packets reach the sink. Thus the average energy consumption is less. However, the packet delivery ratio (PDR) of BECDA is less than TTCDA and EECDA. This is because the additive and divisible aggregation functions are performed at the cluster level as a result the packet delivery ratio of BECDA is less and correspondingly residual energy is more. Thus the network lifetime is also longer for BECDA in comparison with EECDA. Thus, it can be concluded that BECDA is bandwidth efficient and is superior in terms of energy saving if it is compared with EECDA. It is analyzed that the usage of the mobile sink is not seen. As the role of the mobile sink is not explained in any of the phases of the algorithm. Hence, fix sink can also generate the same results instead of mobile sink.

2.17 ENERGY AWARE MULTIHOP MULTIPATH HIERARCHICAL (EAMMH) ALGORITHM

In [33] Rana hameed et.al. in 2015 discussed an energy aware multihop multipath hierarchical (EAMMH) routing protocol. This protocol works in three phases: Setup phase, data transmission phase and periodic phase. In setup phase, once the nodes are deployed then neighbouring nodes are discovered using messages. Then clusters are formed, after that every node will decide whether it will become a cluster head for this round or not. The cluster head election procedure is same as that of LEACH [30]. There are mainly two operations that are performed in setup phase: cluster head election and cluster formation. The second phase of EAMMH protocol is data transmission phase in which nodes are assigned with time slots to transfer the information. The nodes send their sensed information only during their allotted time slot. Nodes receives the data from the neighbouring nodes and then aggregates the data. The aggregated data is send via the optimal path selected from the routing table. The optimal path selection is performed on the basis of heuristic function. The heuristic function depend upon the average energy and minimum hop count. In the third phase, after some time the information in the routing tables's value gets old and this leads to wrong decisions. So the routing table should be updated after sometime. The simulation results shown by the authors conclude that LEACH performs better than EAMMH in small networks. While for large networks EAMMH performs better.

2.18 CLUSTERING AND PRINCIPAL COMPONENT ANALYSIS (PCA) BASED ALGORITHMS

Jun Li et al. [21] in 2016 discussed the two algorithms: the distributed clustering algorithm with data similarity, distributed clustering algorithm based on the principal component analysis (PCA) [51]. In the distributed clustering based algorithm with data similarity, each sensor node is assigned different clusters depending upon the data similarity between the sensor nodes. The data similarity of the different sensor nodes is computed based upon with its neighbour nodes. In data magnitude similarity, the difference between the measured value and the threshold value for the sensed data by the node is computed. While in data correlation, the two neighbouring nodes data is correlated. If both the data magnitude similarity and the data correlation factors are below the threshold value then only the data of the two neighbouring nodes is said to be similar. After computing the similarity of data between the neighbouring node, a graph will be formed by the nodes. And the nodes with a similarity of data will come under one cluster and a cluster head (CH) is selected based upon the node degree i.e. the node with highest node degree is selected as a cluster head. In the distributed clustering, the algorithm is based on PCA. The authors have applied the compression algorithm (PCA) at CH level. After the CH collects the data from all it's cluster members. It stores them in the form of a matrix. Then the PCA algorithm is applied to this stored data and results in a transformation matrix which is smaller than the original stored matrix. This matrix is termed as Low projection matrix, which is then sent to the sink node by the CH. The authors compare the results of their clustering algorithm with that of the LEACH [52] algorithm, k means [53]. The results show that once the clustering is done and PCA is applied, the energy consumption reduces for the data transmission. Finally, the clustering algorithm with PCA provides less energy consumption for the data transmission in comparison to LEACH and the k means clustering algorithm.

2.19 PRIVACY PRESERVING SECURE IN-NETWORK DATA AGGREGATION (PPSDA) ALGORITHM

One of the major issues in WSN is providing privacy to the data i.e protecting crucial information. The Privacy preservation scheme used in [54]; [55]; [56] provides an end to end confidentiality to the data with the minimum energy consumption. In these algorithms, the data is decrypted at the intermediate nodes (aggregation), and it is again encrypted before sending the same to the next node. This may result in losing the privacy of the data as the intermediate node may be an attacker node. So, to provide an end to end security of the sensed data Vishal et al. [18] in 2016 proposed a PPSDA algorithm as shown with the help of flowchart in 2.10. They have used the property of Paillier crypto system [57]. This system uses a probabilistic asymmetric algorithm for performing public key cryptography. In PPSDA scheme, the node first sensed the environmental data in which it is deployed. The sensed data is multiplied with a given threshold and is appended before the sensed data. The appended data is encrypted by the leaf node. All the intermediate nodes can encrypt the data but only the sink node can decrypt the data. This is because of the homomorphic property offered by Paillier crypto system that only allows the addition of data at different intermediate nodes. The sink node decrypts the data.

The performance analysis of PPSDA is performed by comparing it with Secure data aggregation protocol (SDAP) [58] and Secure end-to-end data aggregation (SEEDA) [59]. The results proved that PPSDA performs better than SDAP and SEEDA in term of stability period which further leads to an increase in the lifetime of the PPSDA in comparison to the SDAP and SEEDA. PPSDA has higher data transmission rate in comparison to both the comparative algorithms. This is because homomorphic encryption is used, which provide privacy to the data and also the data aggregation is done with minimum energy consumption. The network traffic analysis of PPSDA is better than the SDAP and SEEDA because in each round more packets are handled by the PPSDA. During the data transmission, less packets are lost in PPSDA so its efficiency is better than SDAP and SEEDA. The accuracy of PPSDA is also good. In SDAP, divide and conquer and commit-and- attest principles are used. Which help to provide the security to the data, hop by hop, but in PPSDA, an end to end security is provided to data and another

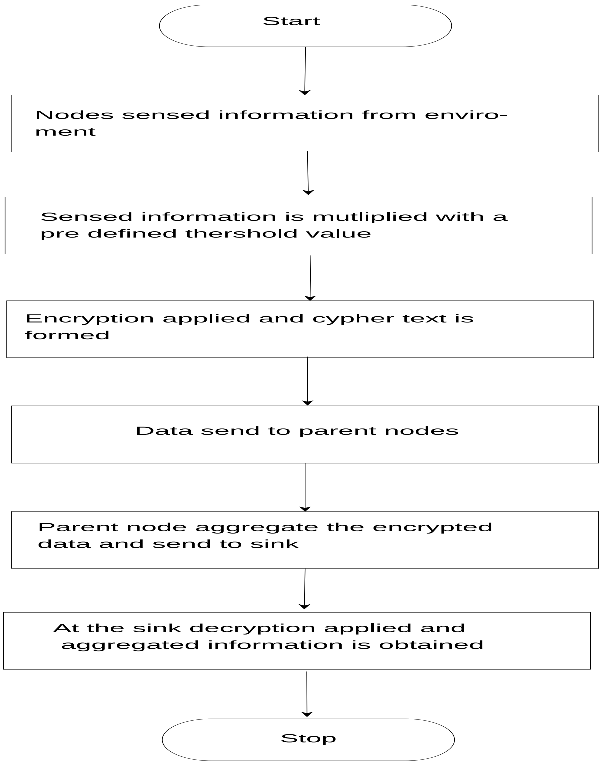


Figure 2.10: Flow chart of PPSDA algorithm

important aspect of PPSDA is that once the source provides an encryption to data then it is only decrypted at the sink. So, PPSDA is more secure than SDAP. Hence, the proposed algorithm is better than the traditional algorithms.

2.20 DISTRIBUTED DATA AGGREGATION SCHEDULING (DAS) ALGORITHM

Sain Saginbekov and Arshad Jhunk [17] in 2017 describes two data aggregation in algorithms out of which one is the distributed aggregation scheduling (DAS) and the another is energy efficient collision (EECF) DAS algorithm. The main goal of these algorithms is to connect two sinks in such a way that the tree formation should be balanced as well as cost effective i.e. Steiner tree problem [60]. The DAS is further categorized into two type of scheduling, weak scheduling for distributed aggregation scheduling (DAS) in which it is considered that node should be DAS labeled with respect to a single path and strong scheduling for DAS in which it is considered that node should be DAS labeled with respect to every path. The authors have mainly focus on weak DAS algorithm. This algorithm includes three phases. In the first phase, the shortest path between the two sinks is computed with the help of sensor node and all the sensor nodes which are in the shortest path are termed as virtual sinks. In the second phase, Greedy Incremental tree (GIT) [16] is used to form a tree between the virtual sinks and sensor nodes towards both the sinks. The extended GIT algorithm is named as balanced tree formation (BTF) algorithm. A node can select the other node as its parent node leaving the current parent node if and only if the other parent node has less hop distance as compared to the current parent node. If in case, the other parent node and the current parent node both are at equal hop distance from the virtual sink then the other parent node can be selected as its parent node if it has less number of children nodes. In the distributed energy balancing algorithm, the authors have used an energy efficient collision free (EECF) Algorithm for a balanced tree. This algorithm is an extension to the weak DAS. Here, the obtained balanced tree from weak DAS is used and the slots are assigned to each child node contiguously in order to avoid the unnecessary wastage of energy for collecting the data from all the children nodes. The performance of EECF DAS algorithm is compared with a cluster based DAS (CDAS) [61]. In order to use two sinks for CDAS algorithm the authors have adapted two ways:

1. Execute CDAS two times i.e. one for each of the sink and is named as 2DAS.
2. Execute shortest path algorithm between the two sinks to form the shortest path the authors have used virtual sinks as a root for the dominating tree [62] and is named SP-DAS.

The results are obtained while keeping sinks at the diagonal position, it is observed that EECF DAS and SP-DAS offers lesser delay than 2DAS. This is because of less delay in slot assignment to the node moreover as the transmission range increases the EECF DAS algorithm performs better but SP-DAS is better for smaller transmission range. In terms of schedule profile, EECF DAS also performs better because the number of the transaction from the sleep state to active is lesser in comparison to SP-DAS and 2DAS. As in EECF DAS, authors balance a tree and assign a contiguous slot, this allows to increase the sleep time as well as it reduces the transaction between active/sleep. This further reduces the energy consumption of the nodes and as the result the transaction energy cost of EECF DAS is reduced. Thus EECF DAS algorithm performs better than SP-DAS and 2DAS.

2.21 OPTIMAL CLUSTERING IN CIRCULAR NETWORKS (OCCN) ALGORITHM

To reduce the energy consumption and to increase the network lifetime, Mahdi Arghavani et al. [11] in 2017 introduced a clustering based algorithm termed as optimal clustering in circular networks (OCCN). In OCCN, initially, the area is partitioned in the form of rings around the sink in a circular fashion. The gap between two rings should be equal to the optimal distance between two nodes. In each ring, there is a certain number of clusters that are made by sink node. To minimize the energy and the time consumed during cluster head selection, every node is allocated time for which it will act as the cluster head. Actually, there is a window of time slots from which each node takes randomly a time slot. This process is called as reservation phase in OCCN algorithm. Then each node sent its reserved slot information along with their Id within their communication distance.

Every node that will receive the same, will compare the distance of other nodes. The node will be selected as cluster head which is near among the nodes from where the reservation message is received. The cluster head's responsibility is to select the specific number of cluster members in a cluster so that there will be load balancing among the clusters. The performance analysis of OCCN algorithm shows that it perform network lifetime enhancement in comparision to LEACH [30] protocol and Hybrid Energy-Efficient Distributed clustering (HEED) [63].

2.22 DYNAMIC ADAPTIVE HIERARCHICAL DATA AGGREGATION (DAHDA) ALGORITHM

Sukhchandan Randhawa et al. [13] has introduced a DAHDA algorithm. The algorithm is described in four different phases. The first phase is location information propagation phase, in which the nodes will send their location information to the sink. The second phase is network partitioning phase, in which base station divides the whole network into four quadrants so that connectivity and coverage can be provided. The third phase is cluster formation phase, in which Meta-Heuristic computational-density based clustering (MHC-DC) [64] is described for cluster formation. Moreover, it is based on the density based clustering (DC) [65] algorithm. The fourth phase is weight assignment phase in which weights are assigned to each node on the basis of their neighbour density. The nodes having lesser weights are selected as data sending nodes. Which has the responsibility to send the data to their respective cluster head. The cluster reformation is done for the scenario where the number of nodes in a cluster is less than the minimum required. The cluster reformation is done with help of MHC-DC.

Furthermore, the authors have optimized DAHDA algorithm and introduced an extended version of DAHDA. In EDAHDA the number of data transmitting nodes are reduced in order to save the energy of the network. Furthermore, the authors have also provided the extension to EDAHDA for managing the sudden burst in the data and is named as modified-EDAHDA. Here the data sent by the nodes in each round is tracked and in case sensed data changes then the weight associated with that node will be reduced so as to increase the probability of data transmission towards cluster head. So in nutshell the modified EDAHDA algorithm the nodes will send the data to their respective cluster head only when the sensed information changes.

The performance analysis of DAHDA, EDAHDA, Modified EDAHDA algorithm is performed by comparing it with LEACH [30]. The results show that DAHDA, EDAHDA, Modified EDAHDA provides effective results in terms of network lifetime, residual energy, data accuracy.

2.23 CLUSTER-CHAIN MOBILE AGENT ROUTING (CCMAR) ALGORITHM

The cluster-chain mobile agent routing (CCMAR) [12] was developed by SelvaKumar Sarirekha in 2017. This algorithm was developed to minimize the transmission delay and energy consumption as well as to increase the lifetime of the network. The CCMAR algorithm is broadly classified into three phases. In the first phase, from all the sensor nodes cluster heads are selected, then each CH is allocated a set of nodes that form the cluster. The clustering concept helps in retaining the residual energy of nodes for the longer time period. In second phase, PEGASIS [35] is used to form the chain between sensor nodes so that data aggregation can be performed. Clustering concept helps in retaining the residual energy of nodes for a longer time period. In the third phase, Mobile Agents (MA) are used to reduce the transmission delay as well as MA helps in gathering the data from the cluster head so that energy consumption of cluster head node is reduced.

The results of CCMAR are compared with LEACH [30], PEGASIS [35] and EECP (Energy efficient cluster-chain based protocol) [66]. It is observed that the CCMAR performs better than the existing algorithms in terms of energy consumption, transmission delay, and network lifetime.

2.24 CONFUSION BASED ENERGY-SAVING AND PRIVACY- PRESERVING DATA AGGREGATION (CESPT) ALGORITHM

Zhang Jun et al. [26] in 2017 introduced a CESPT algorithm to overcome the problem of communication overhead and provides privacy to the data. In this algorithm, there are three phases: Aggregation tree formation phase, confusion phase, aggregation phase. In the first phase, the aggregation tree is created with the help of Tiny AGgregation (TAG) [8] algorithm. In the second phase, the fake data is introduced into the real data to make the confusion, from the security point of view. The fake data is similar to the real data the difference is in the sign i.e +ve and −ve. In the third phase, real data and fake data both are aggregated together to reduce the communication overhead in the network. The data aggregation in k time slots i.e the length of time slot provided to each node depend upon the depth of aggregation tree.

The performance analysis of CESPT is performed by comparing it with SCPDA (Secret confusion of privacy-preserving data aggregation) [67] in terms of traffic analysis, security analysis, and accuracy analysis. The results shows that CESPT performed better than SCPDA in all the above mentioned metrics.

2.25 MOBILE AGENT PROTOCOL BASED ENERGY AWARE DATA AGGREGATION (MAPA)

Mohamed El Fissaoui et al. [14] has introduced the concept in which planning of mobile agents (MA) is performed using cluster heads (CH). Instead of traditional way, where the sink node performs the planning of mobile agents. The authors have introduced the algorithm named as MAPA. The MAPA is consist of three different phases. The first phase is the grouping of SNs (sensor nodes) and selection of CHs in which each node of the network compute its impact factor density. The node with higher impact factor density will have a higher probability to become a CH. Once CH is selected, then all the SN's that lies in its range will form a cluster. This process is repeatedly performed till all the sensor nodes of the network are covered in a cluster. The second phase is Itinerary planning based on MST (Minimum Spanning tree) [68] is performed to form a tree between the cluster heads. The MA's dispatching and data gathering is the third phase in which sink dispatches the mobile agents in the network so that they can collect the data from the cluster heads.

The performance analysis of MAPA is analyzed by comparing it with GCF (Global closest first) [69], MAAD (Mobile agent-based directed diffusion algorithm) [70], IEMF (Itinerary energy minimum for first-source-selection) [71], NOID (Near-optimal itinerary design) [72], DSG (Directional source grouping) [73]. The results show that MAPA algorithm provides considerable results in comparison to earlier discussed algorithms in terms of energy consumption, time consumption.

2.26 SCALABLE AND LOAD BALANCED MOBILE-AGENTS BASED DATA AGGREGATION (SLMADA)

Govind P. Gupta et al. [15] in 2017 introduced SLMADA protocol. SLMADA is introduced to solve the bloating state problem of MA (Mobile agents). The SLMADA provide consists of three phases. The first phase is network setup phase in which the whole area of the network is divided into concentric zones. It is assumed here that each node knows its own co-ordinates as well as that of the sink node. Now, each node will compute their polar co-ordinates using the known co-ordinates and also each node will compute its Euclidean distance from the sink node. Then, every node in the network transmits a message in its communication range so that nodes will be able to know about their neighbours. The second phase is zone co-ordinator selection phase. A zone co-ordinator is selected from each concentric zone and the selection of the zone co-ordinator is performed by the sink on the basis of maximum energy in comparison to that of threshold energy.

Initially, the sink node will send a MA towards the nodes of a zone. Firstly the node will check whether MA is related to its zone or not. If it is not then it will forward the same toward the next zone. Otherwise, the receiving node of the zone (which receive the MA first) will check that whether its residual energy is more than the other node within its communication radius. If any node within that zone will have more residual energy than that of previous node then that node will be elected as a zone co-ordinator and corresponding packet is the end to the sink regarding the selection of zone co-ordinator. The third step is agent migration process, in which sink node sends a MA for each zone co-ordinator. The role of MA is to collect the data from the entire zone and forwards the data to the next zone so that data will be reached at the sink.

The performance analysis of SLMADA protocol, it is compared with TBID (Tree based Itinerary Design) [74], ILS (Iterated local search) [75]. It is observed that SLMADA provides the appreciable results in comparison to TBID, ILS in terms of average energy consumption, response time, network lifetime.

2.27 SECURE DATA COLLECTION SCHEME BASED ON COMPRESSIVE SENSING (SeDC)

Ping Zhang et al. [24] in 2018 discussed a secure scheme for data aggregation i.e secure data collection scheme based on Compressive sensing (SeDC). This scheme consists of four phase: The first phase is setup phase in which a random matrix and public key of homomorphic encryption are distributed among the nodes. In the second phase, each node sensed the data from the environment and program compressive sensing (CS) [76] and then encrypt the CS data. After the encryption is applied than data is send forward to the sink node through intermediator nodes. In the third phase, each intermediate node performs data aggregation with its own data and forward it towards the sink node. The fourth phase is performed at the sink, where sink receives all the encrypted data and then it aggregates the data in cipher domain. Then sink node decrypt the data using the private key. The authors evaluated the performance of SeDC algorithm on real data set and the result shows that this protocol provides better security with low computation cost.

CHAPTER 3

PROPOSED CENTRALIZED DATA AGGREGATION ALGORITHM

The previous chapter concludes that centralized data aggregation algorithms are more energy efficient in comparison to the distributed data aggregation algorithms. This is due to the fact that in centralized algorithms most of the nodes are communicating their data packets within smaller distances and moreover the data is being aggregated at different aggregator node or cluster head which further saves the energy. The concept of energy saving in centralized data aggregation algorithm motivates our research in the domain. In this chapter, we have proposed a centralized data aggregation algorithm i.e. Levelling and Self Localization based Clustering (LSLC). The explanation of the same algorithm is given in the subsequent sections. Section 3.1 covers the detailed explanation of LSLC algorithm. The detailed explanation is accompanied with different diagrams as well as examples to illustrates our protocol effectively. In the section 3.2 pseudocode of our LSLC algorithm is provided. Section 3.3 provides a flowchart of the LSLC algorithm for better and easier understanding the algorithm.

3.1 LEVELLING AND SELF LOCALIZATION BASED CLUSTERING (LSLC) ALGORITHM

In this algorithm, we have made two levels of assumptions: one is at the node level and another is at the network level.

1. Node level assumptions:
    - Node is randomly deployed.
    - All the nodes are homogeneous and static.
    - Data generation rate of every node is same.
    - Nodes are not aware of their geographical location or nor they are deployed with a global positioning system (GPS).

Mandeep Kaur, Amit Munjal, *"Levelling and Self Localization based Clustering algorithm (LSLC) algorithm for Data Aggregation in Wireless Sensor Network"*, Wireless Personal Communication, Springer, [SCI, IF 1.2] [Communicated]

2. Network level assumptions:
    - Base station (B.S), sensor nodes, cluster heads are communicating in full duplex mode.
    - There is one fixed sink in the network.
    - Radio channel is assumed to be symmetric. It means energy required to transmit data from C to D is same that required from D to C.
    - Due to data aggregation, message overheads in the network is assumed to be less.

Assume that N number of nodes are randomly deployed in a square region of $l \times lmetre^2$. In our work, we assume a radio energy model [21], as shown in Figure 3.1 in which we assume that each node has t bits to transmit to the base station.

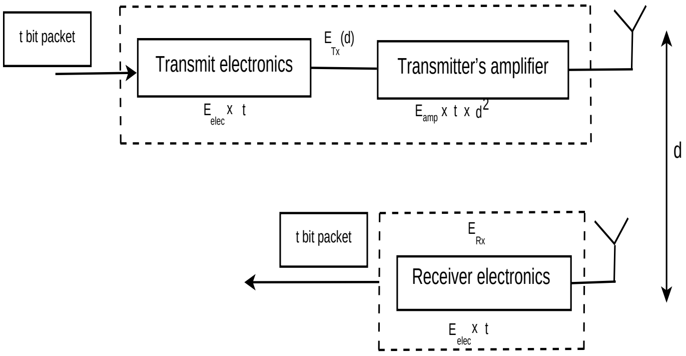


Figure 3.1: First order radio model [21]

The amount of energy consumption required to transmit t bits to the base station is computed using Equation (3.1).

$$E_{Tx}(t, d_{is}) = \begin{cases} t * E_{elec} + t * E_{fs} * d_{is}^2 & \text{if } d_{is} < d_o \\ t * E_{elec} + t * E_{amp} * d_{is}^4 & \text{if } d_{is} \geq d_o \end{cases} \tag{3.1}$$

threshold for d_o is,

$$d_o = \sqrt{\frac{E_{fs}}{E_{amp}}}$$

$$E_{Rx}(t) = E_{elec} * t$$

Where, d_{is} is the distance between transmitter and receiver, d_o is the communication radius of the node, E_{elec} is energy consumption of the circuit per bit by the transmitter and the receiver, $E_{fs} * d_{is}^2$ is the energy consumption of the amplifier in the communication range, $E_{amp} * d_{is}^4$ is the energy consumption of the amplifier when distance is increased from its communication range.

There is no mobility within the network and moreover, the sink is also at a fixed position. Initially, all the nodes have no knowledge about their position because no GPS is used by the nodes. The proposed LSLC algorithm is a centralized algorithm and has a hierarchical based cluster network. The network is 'event driven' i.e node senses the data from the environment whenever some event occurs in the network. The LSLC algorithm is composed of four different phases as shown in Figure 3.2

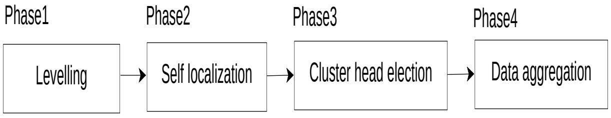


Figure 3.2: Different phases of LSLC algorithm

3.1.1 Leveling Phase

In this phase the base station will broadcast a *HELLO_1* message with minimum power level P_1. All the nodes that listen *HELLO_1* message will set their level as 1 as shown in Figure 3.3 and will not listen to other subsequent *HELLO* messages.

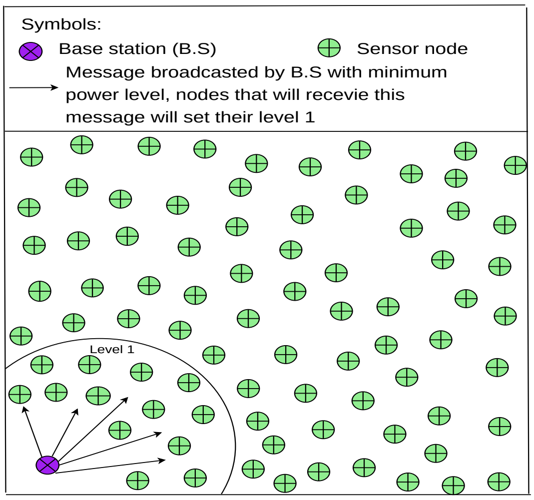


Figure 3.3: Random deployment and leveling of nodes

The *HELLO* message, in general, contains two arguments: the transmission power level and the level number that is to be assigned to the node as shown in Figure 3.4.

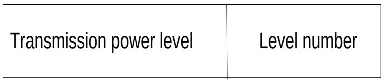


Figure 3.4: *HELLO* message format

3.1.2 Self localization Phase

The sensor nodes of level 1 will start self-localization procedure. As the nodes set their level as level 1 then they will predict their distance from the base station using two-ray ground reflection model [77]. This model is one of the theoretical models which is used to compute the distance between nodes on the basis of received signal strength indicator (RSSI). In this model, both the LOS (line of sight) and NLOS (non-line of sight) path are considered for the pair of transmitter and receiver. The power received by the receiver can be expressed using the Equation (3.2)

$$P_r(d) = P_t * G_t * G_r * \frac{h_t^2 * h_r^2}{d^n} \tag{3.2}$$

where, P_r is the power received by the receiver, P_t is the power transmitted by the transmitter, G_r is the gain of the receiver antenna, G_t is the gain of the transmitting antenna, h_t is the height of the transmitting antenna, h_r is the height of the receiving antenna, d is the distance between transmitter and receiver and n is the path loss exponent, n is 2 for two-ray ground reflection model and n is 4 for the multipath model.

Every node of level 1 will compute its own distance from the base station and store it in its memory. Afterwards level 1 nodes need to know their neighbours and their distances. This is required to apply triangulation method [78] for self-localizing the wireless nodes. Each of the level 1 nodes will communicate in its range a *NEIGHBOUR_SEARCH* message that contains the node ID and the transmit energy level as given in Figure 3.5. After reception of *NEIGHBOUR_SEARCH* message, each node will compute its distance from the neighbours using two-ray ground reflection model.

Node ID	Transmit energy level

Figure 3.5: *NEIGHBOUR_SEARCH* message format

Now, the triangulation method is used, in which two nodes are selected randomly and the base station act as third point (reference node). Consider an example as shown in Figure 3.6. Node1 and Node2 are selected randomly and they have computed their distances from the base station as well as the distance between them i.e. d,e,f respectively as presented in Figure 3.7. The base station coordinates are assumed to be 0,0. The triangulation method is used to compute the angles of a triangle formed by the B.S, Node1 and Node2 as shown in Figure 3.7 i.e. $\angle D$, $\angle E$, $\angle F$ respectively.

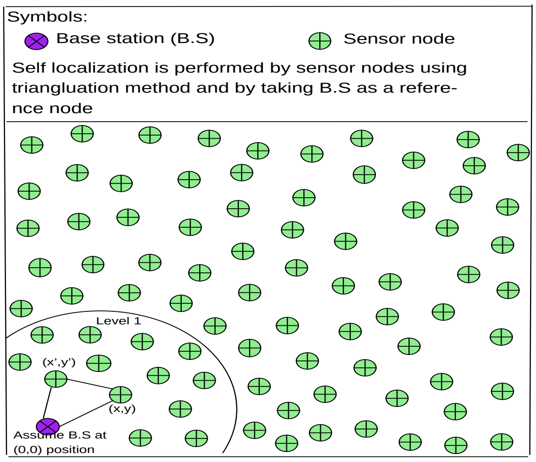


Figure 3.6: Self localization

The two main theorems that are used in triangulation method are Cosine theorem and Sine theorem. The cosine theorem [79] for a given Figure 3.7 is given in Equations (3.3)

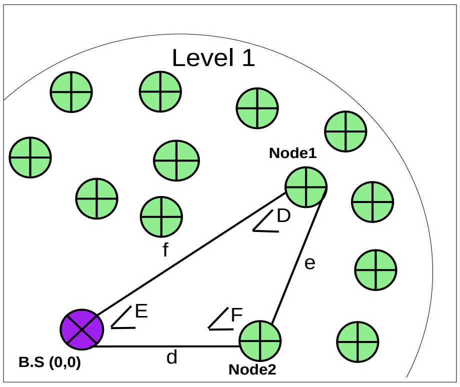


Figure 3.7: Triangulation method

$$
\begin{aligned}
f^2 &= d^2 + e^2 - 2 * d * e * CosF \\
e^2 &= d^2 + f^2 - 2 * d * f * CosE \\
d^2 &= e^2 + f^2 - 2 * e * f * CosD
\end{aligned}
\tag{3.3}
$$

The Sine theorem [79] for a given Figure 3.7 is given in Equation (3.4)

$$\frac{d}{SinD} = \frac{e}{SinE} = \frac{f}{SinF} \tag{3.4}$$

To understand the triangulation method properly let's take an example.

Example: Let's suppose that distance calculated by RSSI between base station (B.S) and

Node1 is f, between B.S and Node2 is d and finally between Node1 and Node2 is e. Lets assume f=14, d=12, e=5 for this example as shown in Figure 3.8.

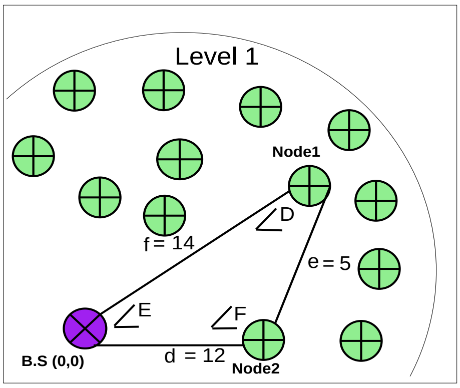


Figure 3.8: Example of triangulation method

Step I: Apply Cosine rule on the side which is largest from all other sides of a triangle that is formed between B.S, Node1, Node2.

$$f^2 = d^2 + e^2 - 2 * d * e * CosF$$

f=14,d=12,e=5

$$14^2 = 12^2 + 5^2 - 2 * 12 * 5 * CosF$$

$$196 = 144 + 25 - (120 * CosF)$$

$$196 - 144 - 25 = -120 * CosF$$

$$27 = -120 * CosF$$

$$\cos^{-1}(\frac{-27}{120}) = F$$

$$F = 103.0028^\circ$$

Step II: Apply Sine rule on any side

$$\frac{e}{SinE} = \frac{f}{SinF}$$

$$\frac{5}{SinE} = \frac{14}{Sin(103.0028)^\circ}$$

$$\frac{5}{SinE} = \frac{14}{.9743}$$

$$\frac{5}{14.3684} = SinE$$

$$Sin^{-1}(0.34798) = E$$

$$E = 20.36^{\circ}$$

Step III: Now apply the rule that sum of all the angles of a triangle is 180°

$$E + F + D = 180^{\circ}$$

$$20.36^{\circ} + 103.0028^{\circ} + D = 180^{\circ}$$

$$123.3628^{\circ} + D = 180^{\circ}$$

$$D = 56.64^{\circ}$$

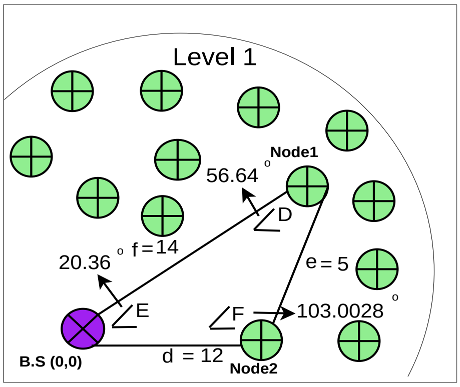


Figure 3.9: Resultant triangle after triangulation method

Figure 3.9 shows all the angles of a triangle obtained using triangulation method. Now we need to find the co-ordinates of Node 1 and Node 2 by assuming that B.S is at (0,0) position as shown Figure 3.9. The co-ordinates of Node 2 can be easily found just by adding value of d (i.e. 12) to the x co-ordinate of Node 2 and y co-ordinate is not varying so it will be same as earlier. Thus co-ordinate of Node 2 will become (12,0). For finding the co-ordinates of Node 1 distance formula is used, which is taken from [80] and given in Equation (3.5).

$$Distance\ between\ B.S\ and\ Node\ 1 = \sqrt{(x\_Node1 - x\_B.S)^2 + (y\_Node1 - y\_B.S)^2} \tag{3.5}$$

$$Distance\ between\ Node\ 1\ and\ Node\ 2 = \sqrt{(x\_Node1 - x\_Node2)^2 + (y\_Node1 - y\_Node2)^2} \tag{3.6}$$

Where, x_B.S is the x co-ordinate of B.S i.e. 0, y_B.S is the y co-ordinate of B.S i.e. 0, x_Node 2 is the x co-ordinate of Node 2 i.e. 12, y_Node 2 is the y co-ordinate of Node 2 i.e. 0, x_Node 1 is the x co-ordinate of Node 1 i.e. to be find, y_Node 1 is the y co-ordinate of Node 1 i.e. to be find. Distance between B.S and Node 1 i.e. f= 14 Distance between Node 1 and Node 2 i.e. e= 5

Substitute all the values in Equation in (3.5, 3.6).

$$\sqrt{(x\_Node1-0)^2+(y\_Node1-0)^2}=14$$

$$\sqrt{(x\_Node1)^2+(y\_Node1)^2}=14$$

Squaring both the sides

$$(x\_Node1)^2+(y\_Node1)^2=196 \tag{3.7}$$

$$\sqrt{(x\_Node1-12)^2+(y\_Node1-0)^2}=5$$

$$\sqrt{(x\_Node1-12)^2+(y\_Node1)^2}=5$$

Squaring both the sides

$$(x\_Node1-12)^2+(y\_Node1)^2=25 \tag{3.8}$$

Comparing both the Equation (3.7, 3.8) by changing the sign of Equation (3.8)

$$(x\_Node1)^2-[(x\_Node1)^2+144-2*x\_Node1*12]=196-25$$

$$(x\_Node1)^2-(x\_Node1)^2-144+24*x\_Node1=171$$

$$-144+24*x\_Node1=171$$

$$24*x\_Node1=171+144$$

$$24*x\_Node1=315$$

$$x\_Node1=\frac{315}{24}=13.125$$

$$x\_Node1=13.125$$

Now, substitute the value of x_Node 1 in Equation (3.7)

$$(13.125)^2+(y\_Node1)^2=196$$

$$172.26+(y\_Node1)^2=196$$

$$(y\_Node1)^2=196-172.26$$

$$(y\_Node1)^2 = 23.74$$

$$y\_Node1 = 4.87$$

Hence, co-ordinates of Node 1 is (13.125,4.87). All the calculated co-ordinates of three nodes are shown in Figure 3.10.

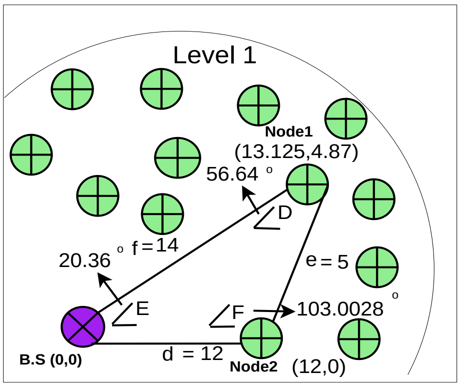


Figure 3.10: Co-ordinates of nodes using triangulation method

In this way co-ordinate's of the level 1 nodes are calculated and store its location (co-ordinates) in their memories and further these nodes will act as the reference nodes for other nodes in the network for computing their co-ordinates.

3.1.3 Cluster head formation Phase

In the third phase, each node will broadcast the *ND_NEIGHBOUR* message in their communication radius (R). This message contains three attributes i.e. node ID, residual energy and its geographical location as given in Figure 3.11. All the nodes will wait for the expiry of the timer to receive the *ND_NEIGHBOUR* message from other nodes. Every node needs to maintain a *NEIGHBOUR_NODE_INFORMATION_TABLE* that contain node ID, its location and residual energy of every neighbour nodes as given in Figure 3.12.

Node ID	Residual energy	Geographical location

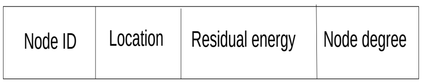
Figure 3.11: *ND_NEIGHBOUR* message format

Node ID	Geographical location	Residual energy

Figure 3.12: *NEIGHBOUR_NODE_INFORMATION_TABLE* format

After the expiry of the timer, each node will compute its node degree, simply by calculating the number of nodes present in its own *NEIGHBOUR_NODE_INFORMATION_TABLE*.

Now, each node will communicate its node degree to all its neighbours present in their *NEIGHBOUR_NODE_INFORMATION_TABLE* and update their *TABLE* by adding one more parameter i.e. node degree as shown in Figure 3.13 and also all the nodes (non-CHs, CH) will periodically update their neighbour node information table.



Node ID	Location	Residual energy	Node degree

Figure 3.13: Updated *NEIGHBOUR_NODE_INFORMATION_TABLE* format

Now each node will compare its residual energy, node degree with all the other nodes present in its *NEIGHBOUR_NODE_INFORMATION_TABLE*. If the node has higher node degree as well as higher residual energy among its competent nodes then it will elect itself as a cluster head for that round, otherwise, the node will not be cluster head. The elected cluster head will forward the *SELECTED_CLUSTER* message to all its neighbours and at the same time, it will also send the message to the base station as presented in Figure 3.14. The cluster head will wait for a negative acknowledgement (NACK) till the expiry of the timer from the neighbour nodes. If the cluster head receives any NACK it means there is another node which comes under two or more cluster heads. In this scenario, the node has chosen the cluster head which is nearer and hence provide NACK to the other cluster heads. In order to avoid the collision of packets send via non-CH nodes, the cluster head will provide TDMA (Time division multiple access) slots to all the non-CH nodes within its communication radius.

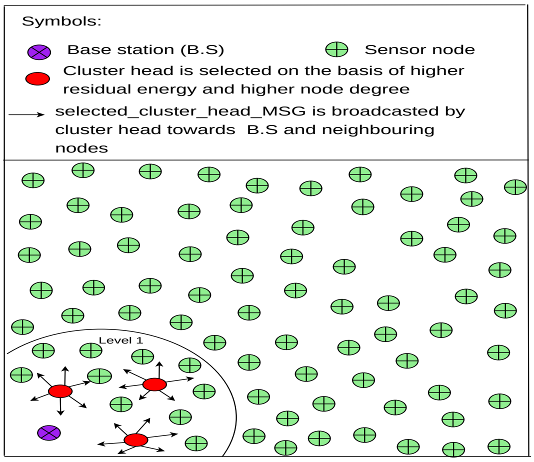


Figure 3.14: Selected cluster head broadcast a message

The *SELECTED_CLUSTER* message contains node ID, cluster head's level, its location and residual energy of the respective cluster head, as given in Figure 3.15.

The base station will maintain a *CLUSTER_HEAD_INFORMATION_TABLE* to store the

Node ID	Cluster head's level	Location	Residual energy

Figure 3.15: *SELECTED_CLUSTER* message format

information about all the cluster heads. This table contains: node Id, cluster head's level, location, residual energy as shown in Figure 3.16.

Node ID	Cluster head's level	Location	Residual energy

Figure 3.16: *CLUSTER_HEAD_INFORMATION_TABLE* format

When the first level is formed then the base station will increase its power level and again broadcast a *HELLO_2* message. The nodes that receives this message will set their level as level 2. The nodes of level 2 will then start their localization phase. The nodes of level 2 will take the reference point of level 1 (upper level) nodes to find their location. After localization phase nodes of level 1 will perform the cluster head election phase. Once the cluster head is elected then cluster heads of level 2 will inform the base station by sending the *SELECTED_CLUSTER* message. When the base station receives the message from the cluster head of level 2 then the B.S will increase its power in steps i.e. P_3, P_4, P_5 and so on, the nodes will perform localization, cluster head election repeatedly for every level till entire coverage area is covered as shown in Figure 3.17.

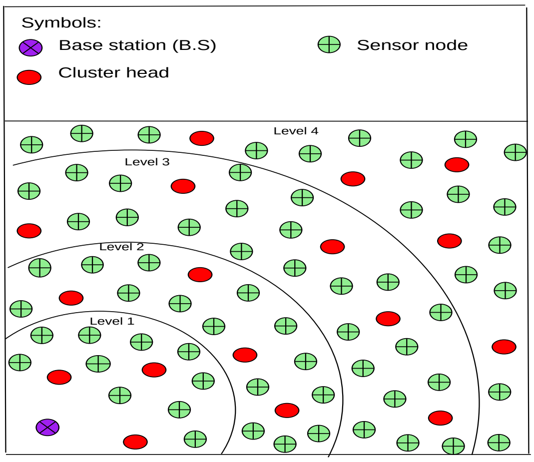


Figure 3.17: Different levels and cluster heads

Once the levelling, self-localization and cluster head formation phases are completed then the base station will broadcast the *CLUSTER_HEAD_INFORMATION_TABLE* which contain node ID, cluster_head's_level, its location as well as it's residual energy. Every cluster head will maintain a *CLUSTER_HEAD_INFORMATION_TABLE* and choose the nearest cluster head for multihop communication towards the B.S.

3.1.4 Data Aggregation phase

The data aggregation is performed by CH's of different levels whereby all the non-CH nodes will sense the information and send the same to their respective CH's. The energy consumed by cluster head is discussed as follows using energy consumption model, let's suppose that there are N nodes that are randomly distributed in $l \times lmetre^2$ area. Suppose that in the network their exist c number of clusters. On an average, there will be $\frac{N}{c}$ number of nodes per cluster. In the cluster there will x cluster head (CH) and $(\frac{N}{c} - x)$ are non-CH's (nodes). The energy consumed by a cluster depends upon two type of energy consumption: one is the energy consumed by CH's and another is energy consumed by non-CH's nodes. The Energy consumed by non-CH nodes while transferring the t bits to the CH is given in Equation (3.9) [81]

$$E_{non-CH} = t.E_{elec} + E_{amp}(t, d) \tag{3.9}$$

The energy consumed by CH is computed by performing the sum of three energies as Energy consumed while receiving the data from the non-CH's, energy consumed while aggregating the data and energy consumed when aggregated data is being transferred to the base station it is given by equation 3.10

$$E_{CH} = (\frac{N}{c} - m).t.E_{elec} + \frac{N}{c}.t.E_{DA} + E_{TxtoBS}(t, d) \tag{3.10}$$

Hence, total energy consumed in a cluster is given in equation 3.11

$$E_c = E_{CH} + (\frac{N}{c} - m).E_{non-CH} \tag{3.11}$$

Consider a scenario in which CH of level 1 is having four non-CH members. Suppose these non-CH nodes are sensing temperature and humidity of a particular area. The CH of level 1 will receive the information from all its non-CH nodes and is summarized in Table 3.1. The CH node will add its own data to the collected data and perform the *MEAN* as shown in Figure 3.18.

$$Mean(Temp.) = (14.4 + 14.2 + 14.0 + 14.8 + 13.9) \div 5 = 14.26 \tag{3.12}$$

$$Mean(Humidity) = (2.2 + 2.4 + 3 + 1.9 + 2) \div 5 = 2.3 \tag{3.13}$$

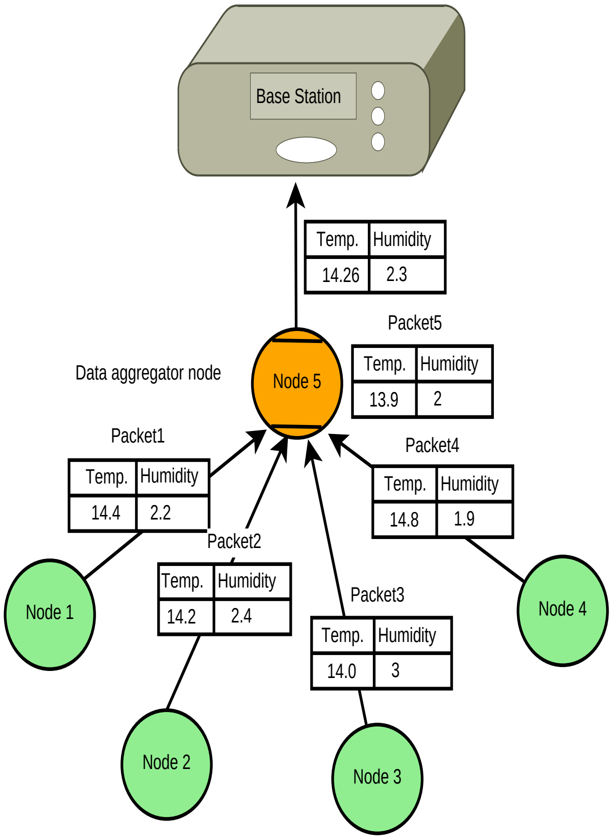


Figure 3.18: Example of data aggregation tree

Table 3.1: Data collected by nodes

Sr No	**Node**	**Functions**		**Packet**
		Temp.	**Humidity**	
1	Node1	14.4	2.2	Packet1
2	Node2	14.2	2.4	Packet2
3	Node3	14.0	3	Packet3
4	Node4	14.8	1.9	Packet4
5	Node5	13.9	2	Packet5

Due to data aggregation being performed by CH (i.e. Node5), only one packet is transmitted towards the base station instead of five packets (in case of data aggregation was not performed). The packet contains mean of temperature and humidity as given by Equation (3.12 and 3.13). Hence, we can conclude that due to data aggregation, the number of packets transmitted is less, hence the bandwidth is saved and moreover less energy is used to send packets and thereby the energy of the node will be saved. Hence, the lifetime of the network will be enhanced. The CH will send the aggregated data to the B.S via single hop or multiple hops depending upon its distance from the B.S as shown in Figure 3.19.

In this proposed protocol, the CH will continue to remain as CH for consecutive rounds, until it drains its energy below a certain threshold. Once the residual energy

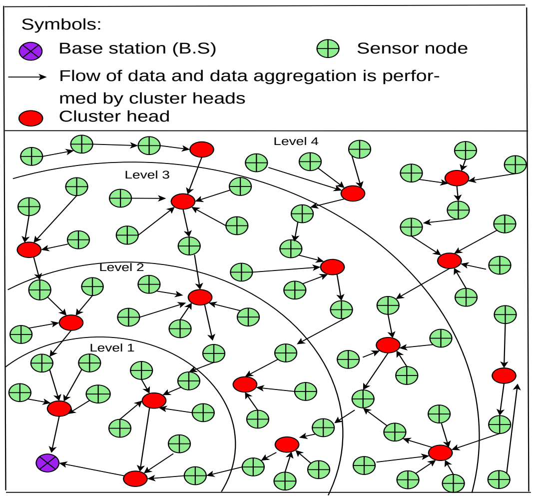


Figure 3.19: Data aggregation

reaches the threshold energy then the cluster head will select a non-CH node from its *NEIGHBOUR_NODE_INFORMATION_TABLE*, (whose node degree as well as residual energy is maximum) as a cluster head. The cluster head then announces the new cluster head message within its cluster as well as to the base station. Now the cluster head will act as the non-CH node for the subsequent rounds. The base station, as well as non-CH nodes, will update their tables.

3.2 PSEUDOCODE OF LSLC ALGORITHM

The pseudocode of the LSLC algorithm is presented in Algorithm 3.1 which is beneficial for quick understanding of our protocol.

Algorithm 3.1 LSLC proposed algorithm

```
Levelling Phase:
Base station (B.S) broadcast a HELLO_1 message with minimum power.
The nodes which receive the HELLO_1 message will set their level as 1.
for each level do
    Self Localization Phase:
    if level==1 then
        Nodes will calculate their distance from B.S using two-ray ground reflection model or multipath model depending upon their distance.
    end if
    Each node of the current level will send a NEIGHBOUR_SEARCH message to all other nodes in its communication range.
    Each node of the level will compute their distance from the neighbouring nodes using two-ray ground reflection model.
    Randomly two nodes are selected.
    if Co-ordinates of both the nodes are unknown then
        Assume third node as B.S. (0,0).
```

```
14:     else if Co-ordinate of one node is known then
15:         Choose any localized or unlocalized node as a third node.
16:     else
17:         Choose any unlocalized node as a third node.
18:     end if
19:     Then apply triangulation method on the selected nodes to compute their co-ordinates.
20:     Repeat step 11 to step 18 until the co-ordinates of all the nodes of the current level are
      calculated.
21:     Cluster head formation phase:
22:     Each node will broadcast ND_NEIGHBOUR message in its communication radius.
23:     Each node will then wait for the expiry of the timer to receive the ND_NEIGHBOUR
      message from other nodes and will store its information in NEIGHBOUR
      _NODE_INFORMATION_TABLE.
24:     All the nodes will periodically update their NEIGHBOUR_NODE_INFORMATION_TABLE.
25:     Each node will communicate its node degree and residual energy to all its neighbouring
      nodes.
26:     Each node will compare its node degree and residual energy with all its
      NEIGHBOUR_NODE_INFORMATION _TABLE.
27:     if neighbouring node has higher node degree as well as residual energy then.
28:         the node will be selected as a cluster head.
29:     else
30:         the node will not be the cluster head.
31:     end if
32:     Cluster head will send the SELECTED_CLUSTER message to all its neighbour and B.S.
33:     If cluster head receives any NACK from its neighbour nodes that means the neighbouring
      node is not a part of the selected cluster head.
34:     Increase the power level to next level and broadcast the HELLO_LEVEL+1 message.
35:     Nodes that will receive the HELLO_LEVEL+1 message will set their level as level+1.
36: end for
37: Base station will broadcast the CLUSTER_HEAD_INFORMATION_TABLE in the network
      which will help cluster heads in performing multi-hop communication.
38: Data aggregation phase:
39: Each node will send information to their respective cluster head.
40: Cluster head will add its own data to the collected data and compute MEAN of the data
41: Cluster head will send the aggregated data to the B.S by using single hop or multi-hop
      communication.
42: Repeat step 37 to step 41 until residual energy is greater than the threshold.
43: if Residual energy is below the threshold value then
44:     Cluster head will select the non-CH node from its NEIGHBOUR _NODE
      _INFORMATION _TABLE as cluster head whose node degree as well as residual energy is
      maximum and go to step 32.
45: end if
```

3.3 FLOW CHART OF LSLC ALGORITHM

The Figure 3.20 shown the flow chart of LSLC protocol. This facilitates to understand the sequence of steps involved in the implementation of a protocol.

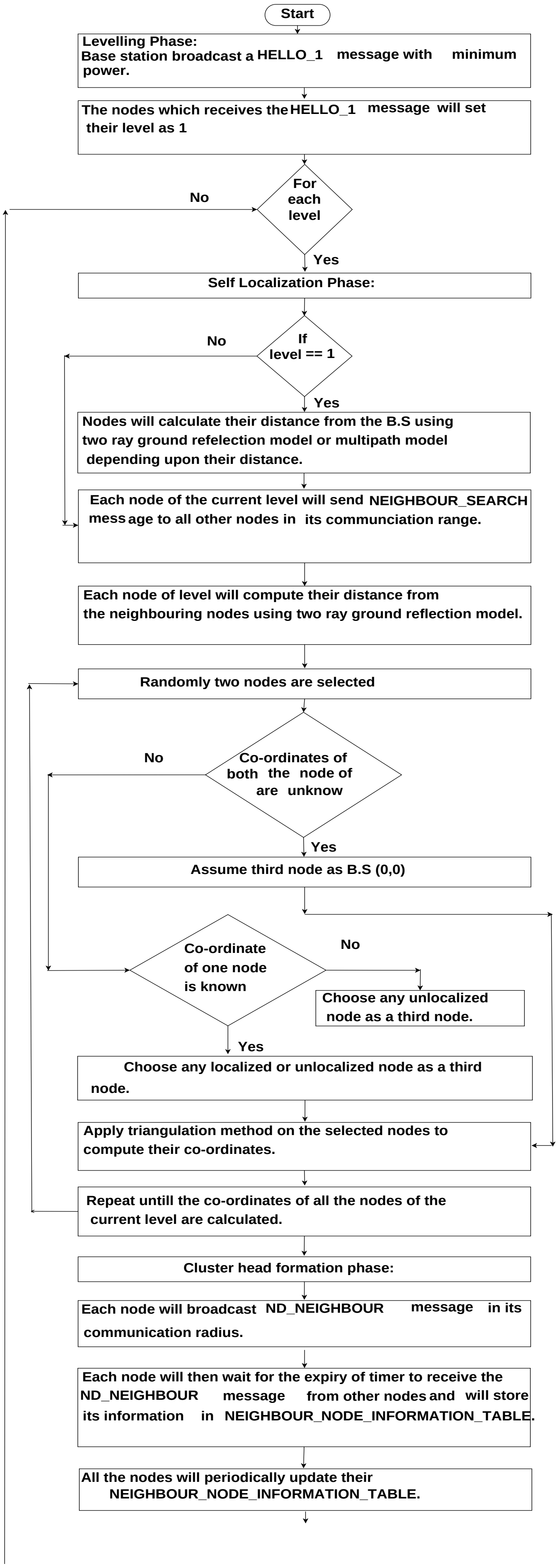


Figure 3.20: Flow diagram of LSLC algorithm

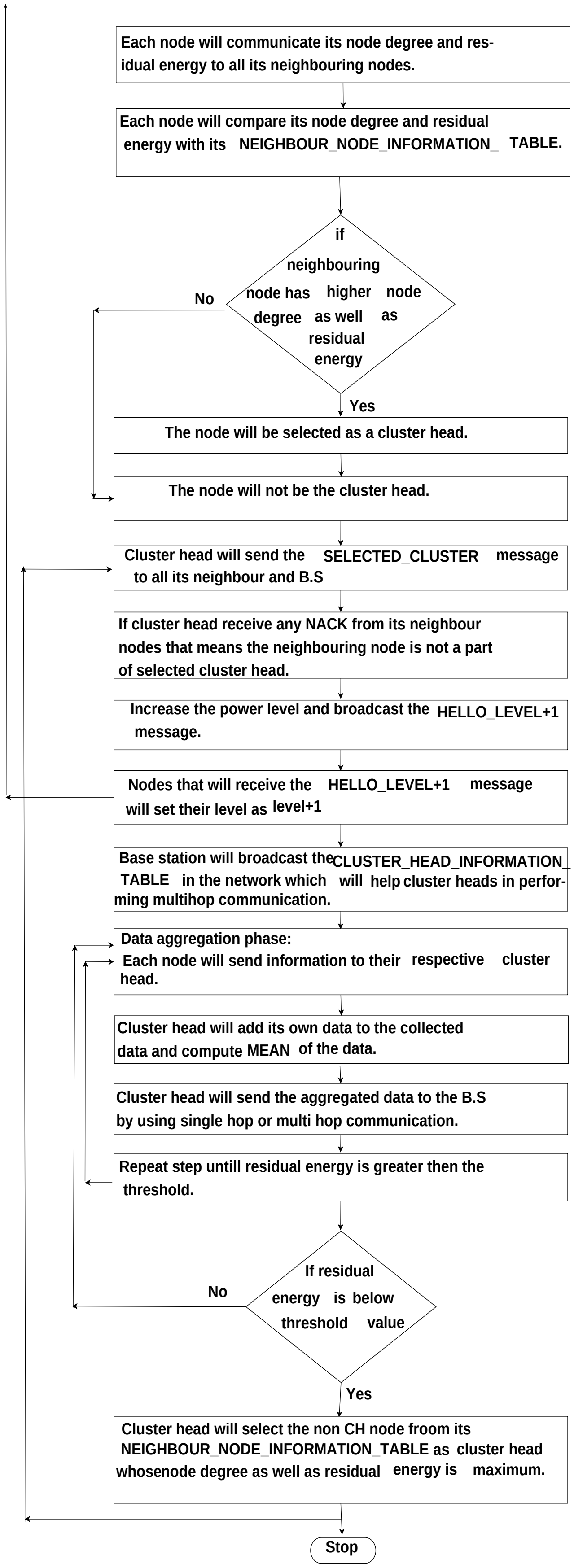


Figure 3.20: Flow diagram of LSLC algorithm

CHAPTER 4

SIMULATION RESULTS

In this chapter, we have presented the simulation results of our purposed LSLC data aggregation algorithm for a set of network parameters. The first half of the chapter covers the different performance evaluation metrics that are being used for the analysis. The second half covers the performance analysis of LSLC algorithm with the existing data aggregation algorithms such as LEACH, MODLEACH, EAMMH, EESAA. Moreover, we have also presented the impact of different performance metrics on LSLC protocol.

4.1 PERFORMANCE EVALUATION METRICS

1. **Network lifetime:** It is defined as a time for which network is performing its desired task effectively for which it is deployed in a particular area of interest. It is always desirable to have higher network lifetime.
2. **Residual energy:** The residual energy is the remaining energy of the nodes after the network is the expired lifetime. It is defined as the ratio of the sum of remaining energy of all the nodes to the total number of nodes.
3. **Latency:** It is defined as the time taken by a packet to travel from the source node to sink node. This latency should be as small as possible for high speed applications.
4. **Packet delivery Ratio (PDR):** It is defined as the ratio of the total number of packets received at the base station to the total number of packets transmitted by all the nodes. Higher packet delivery ratio indicates less loss of packets.
5. **Packet to base station:** It is defined as a measure of the total number of packet received at the base station to the total number of packets generated.

4.2 PERFORMANCE ANALYSIS

The analysis of LSLC algorithm is done by comparing it with some existing algorithms such as LEACH, MODLEACH, EAMMH, EESAA in term of the performance metrics as discussed in the previous section. The simulation parameters are summarized in Table 4.1. For comparison purpose, we have taken the number of nodes as 100 and the initial energy of nodes is 0.5 joule. Moreover, we have also presented the impact of different performance metrics on LSLC protocol.

4.2.1 Comparative analysis of LSLC with existing algorithms

The performance of LSLC protocol is analyzed by comparing it with some existing algorithms on the basis of various parameters such as network lifetime, residual energy, PDR, the packet sent to B.S and Latency. The behavior of LSLC algorithm is analyzed by simulating it on *MATLAB R2016a* with the simulation parameters as shown in Table 4.1. The layout of the network during simulation is shown in Figure 4.1. All the parameters discussed in the previous section are analyzed as follows:

Table 4.1: Simulation parameters

S.No	Parameters	Values
1.	Area of deployment	$100 \times 100m^2$
2.	Number of nodes	100
3.	Position of Base station	$0, 0m$
4.	Length of packets	4000 bits
5.	Initial energy of nodes (E_o)	0.5J
6.	Transmitter, receiver energy per node (E_{elec})	50nJ/bit
7.	Free space energy (E_{fs}) $d \geq d_o$ (cluster to B.S)	$10pJbit/m^2$
8.	Path energy ($Eamp$) $d \leq d_o$ (cluster to B.S)	$0.0013pJ/bit/m^4$
9.	Amplification energy (Intra cluster communication)$d \geq d_1$	$(E_{fs})/10$
10.	Amplification energy (Intra cluster communication) $d \geq d_1$	$(E_{mp})/10$
11.	Data aggregation energy (E_{DA})	5nJ/bit/message
12.	Total rounds	3000

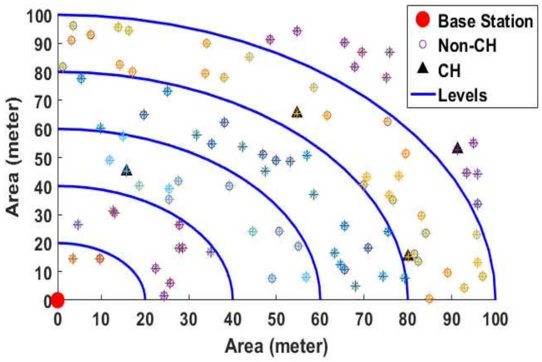
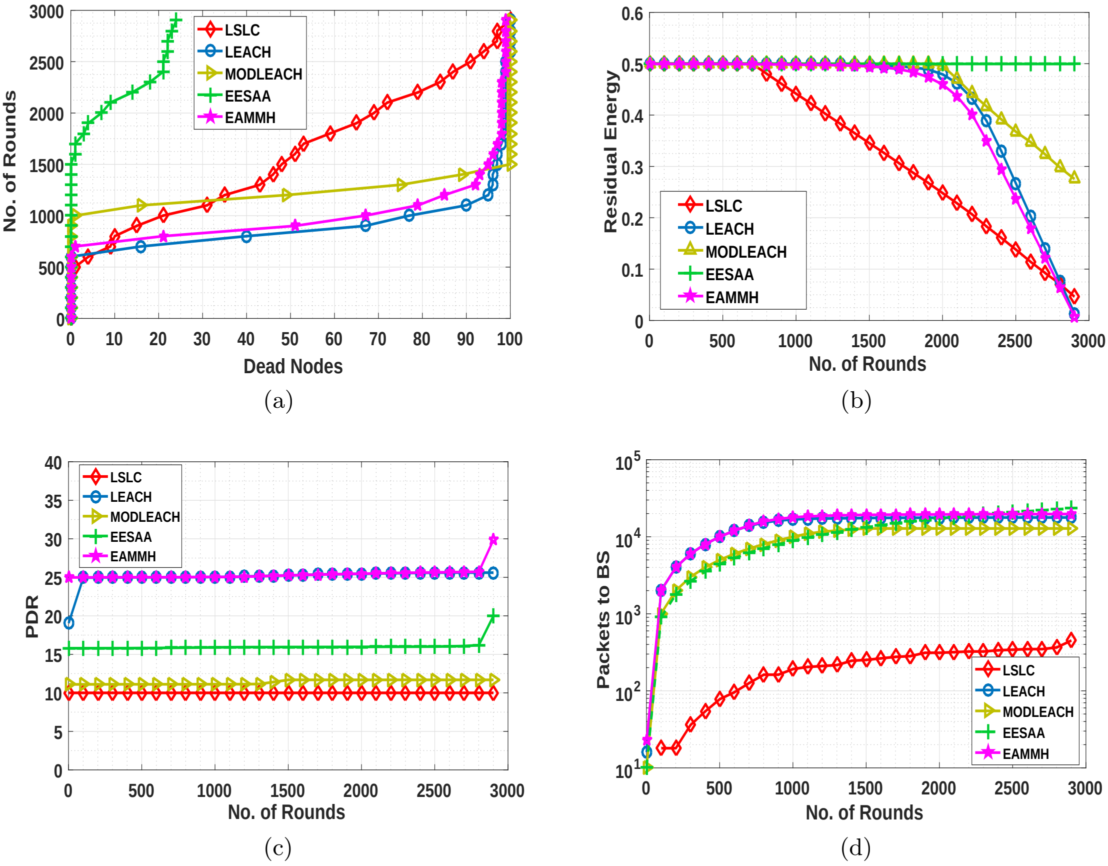


Figure 4.1: Network layout for LSLC

Network lifetime: The simulation result shows that the lifetime of LSLC is better than LEACH [30] and is approximately 56% when the analysis is considered for 90% dead nodes (2500 rounds), as shown in Figure 4.2a. Although MODLEACH [32] and EAMMH [33] performs better than LSLC algorithm for the initial round but if we consider the scenario of 90% dead nodes than LSLC perform approximately (approx.) 48% and 44% better than EAMMH and MODLEACH respectively. The higher lifetime of LSLC algorithm is due to the fact that data aggregation is being performed at different levels by the cluster head and this saves a lot of energy of the nodes. Apart from the LSLC algorithm doesn't perform cluster head reelection for the subsequent rounds. In contrast to it, LSLC offers lesser lifetime than EESAA [31] in which the duty cycling concept of nodes is used.



(a) (b) (c) (d)

Figure 4.2: Comparison with other existing algorithms, (a) Network Lifetime (b) Residual Energy (c) PDR (d) Packet to BS

Residual energy: The Figure 4.2b shows that residual energy of EESAA is highest in comparison with all the other algorithms due to sleep and awake concept. In comparison to other data aggregation algorithms such as: LEACH, EAMMH, MODLEACH. LSLC perform better in term of % residual energy with 48.14%, 41.61% and 62.16% respectively. This reflects that in LSLC all the nodes use their energy efficiently till the network lifetime is over. It means that LSLC is an energy efficient algorithm in comparison to other algorithms.

PDR: The PDR of LSLC algorithm is very less as shown in Figure 4.2c which is approx.

60.78% less than LEACH, 16.66% less than MODLEACH, 37.5% less than EESAA and 60.78% less than EAMMH for 2500 rounds. This is because in LSLC algorithm data aggregation is performed at different levels and this allows fewer packets to be sent to the base station and thereby PDR is decreased.

Packet sent to B.S: Packets sent to B.S is very less in LSLC algorithm as shown in Figure 4.2d. Which is 97.24%, 97 %, 97.34 %, 97.27 % less than LEACH, MODLEACH, EESAA,EAMMH respectively for 2500 round. This is because in LSLC $MEAN$ of the packets are taken by CHs and then forwarded to B.S due to which the number of packets sent to B.S is reduced. This saves the bandwidth of the network and also it helps the nodes to save their energy by sending fewer packets.

Latency: The latency of LSLC algorithm is comparatively much higher than the existing data aggregation algorithm as presented in Figure 4.3. The reason behind higher latency of our algorithm that data aggregation is performed at the multiple levels by the CHs which consumes time and hence increases the latency.

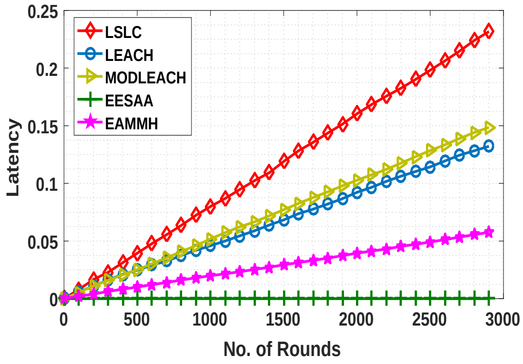


Figure 4.3: Latency

Summary of comparison results

The Table 4.2 summarize the simulation results of LSLC protocol and it also represents the comparisons of results with the existing algorithm. The ↑, ↓ shows the increasing and decreasing trends of different performance metrics of LSLC algorithm w.r.t. other existing algorithms.

The Table 4.2 concludes that the proposed LSLC algorithm provides the considerable improvement in almost all the parameters except latency. Hence, we can say that LSLC protocol is able to enhance the network lifetime and moreover, it is also an energy efficient algorithm. The LSLC protocol saves the bandwidth of the network by sending the aggregated packets instead of single packets. Although the latency of LSLC is increased in comparison to other protocols because aggregator nodes take time to aggregate the data packets. Overall, the LSLC

Table 4.2: Summary of results

Parameters↓	**Existing Algorithms**				**Proposed Algorithm**
	LEACH	MODLEACH	EESAA	EAMMH	LSLC
Network lifetime	56%	44%	-	48%	↑
Residual energy	48.14%	62.16%	72%	41.66%	↓
PDR	60.78%	16.66%	37.5%	60.78%	↓
Packet sent to B.S	97.24%	97%	97.34%	97.27%	↓
Latency	47.5%	35%	98%	77.5%	↑

algorithms provide beneficial results for WSN.

4.2.2 Impact of Performance Metrics on LSLC

In order to study the impact of different performance metrics on LSLC protocol, we have simulated LSLC protocol under different network conditions and the same is discussed in subsequent sections.

4.2.2.1 *Impact on Network lifetime*

This section covers the impact on network lifetime for the LSLC algorithm, with aggregation and without aggregation for a 100 nodes network that is deployed in 100*100m^2 area. The Figure 4.4 illustrates that the network lifetime with data aggregation increases in contrast to that without data aggregation. The Figure 4.4 reveals that approx. 10.71% of network lifetime is enhanced when data aggregation is performed for LSLC protocol in comparison to the scenario of without data aggregation for the same LSLC protocol.

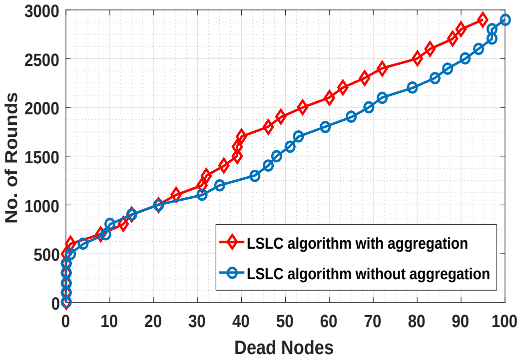


Figure 4.4: Impact on network lifetime

This is due to the reason that in LSLC with data aggregation, aggregation function is applied by aggregator nodes (CH) which results in the reduction of the number of packets sent towards the B.S by removing redundant packets, this concept of aggregation saves a lot of energy of the nodes and also energy of the data aggregator nodes are saved by providing the rotation to the data aggregator nodes when their energy is below a certain threshold value. Then non CH node will be selected as current CH on the basis of certain parameters like node degree, residual energy. On the other hand, LSLC without aggregation means all the packets are sent as it toward the base station instead of aggregating due to which a lot of energy of the nodes is used. Because the nodes which are far away from the base station will drain their energy early and results in the network lifetime decreased as the comparison to with aggregation of LSLC algorithm and also network become less energy efficient.

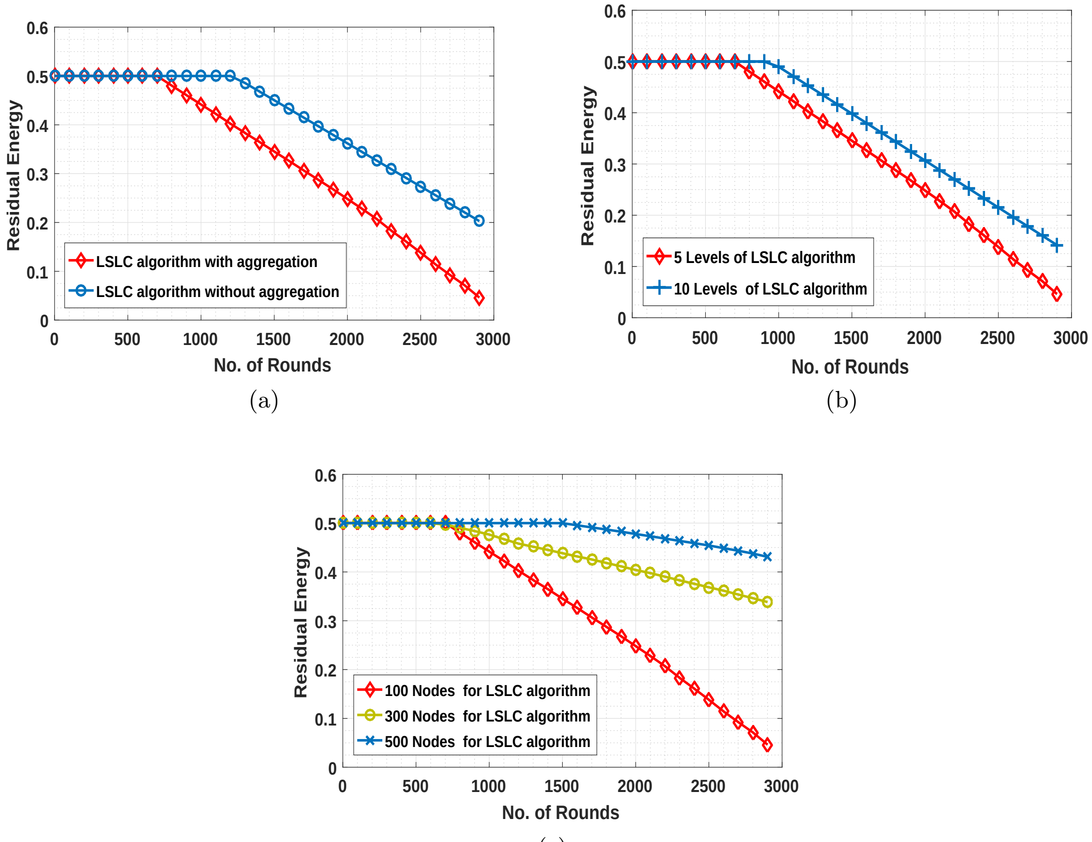
4.2.2.2 *Impact on Residual energy*

In this section we will discuss the impact on residual energy for LSLC protocol for three different scenarios as follows:

- Impact of data aggregation.
- Impact of number of levels.
- Impact of node density.

The Figure 4.5a presents the results of residual energy in LSLC for data aggregation and without data aggregation. The simulation results show that approximately 30% of residual energy is more for the scenarios of no data aggregation. This shows that energy of the nodes is not efficiently being utilized in this scenario. While on the other hand, less residual energy is there when the network uses the concept of data aggregation. This is because nodes sent the data packet to the shorter distance i.e. to nearby cluster head that saves the energy and also the CH keeps on changing at regular intervals so that one CH should not get exhausted. This concept saves the energy of the non-CH nodes and also the CH node. Thus, by data aggregation in LSLC protocol nodes are available for a longer time which enhances the network lifetime. Moreover when the network lifetime is over, then very less residual energy with data aggregation scenario shows the efficient utilization of network energy.

The Figure 4.5b shows the impact of residual energy of LSLC protocol with increasing number of levels. The residual energy for 10 level LSLC algorithm is 45.45% more than that of 5 levels LSLC algorithm. This is due to the fact that as the number of levels is increased from 5 to 10 while keeping the number of nodes and area constant. The number of CH increases while



(a) (b) (c)

Figure 4.5: Impact on Residual energy, (a) With data aggregation and without data aggregation (b) Number of levels (c) Node density

non-CH nodes associated with $CH's$ decrease in comparison to the 5 levels. In 10 level scenario, the inter CH communication increases and thus the CH closer to the base station will die early and thereby leaving higher residual energy when the lifetime expires.

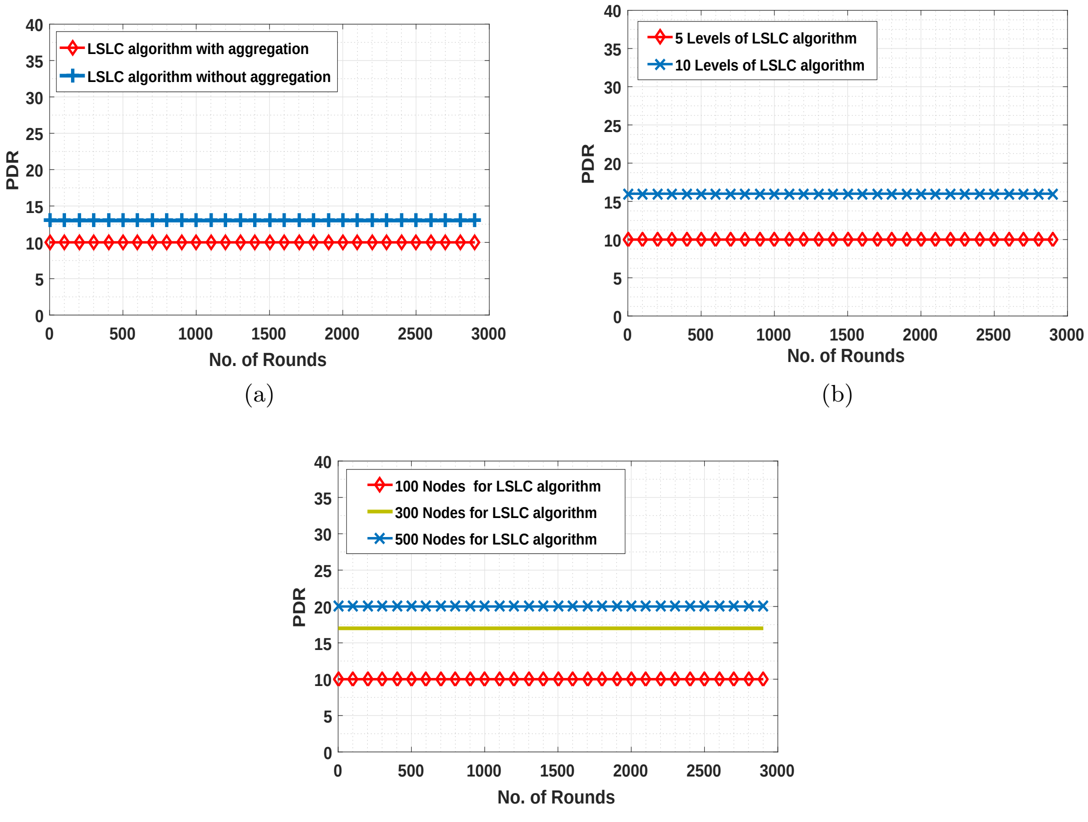
The Figure 4.5c shows the impact of residual energy on node density i.e. node density is increased from 100, 300, 500 nodes while keeping the same area. The Figure 4.5c reveals that as node density is increased, the residual energy is also increased. It is analyzed that w.r.t 100 nodes, the residual energy is increased for 300 nodes and 500 nodes by 68.42%, 72.72% respectively. The reason behind the increase in the residual energy is that as the number of nodes is increased the distance between non-CH and CH is decreased but at the same time CH (data aggregator node) have to collect the more number of packets and for aggregation, so aggregation energy is also increased. It means that CH nodes will consume their energy at a faster rate. Thus when the network dies, more residual energy of nodes shows inefficient energy utilization.

4.2.2.3 *Impact on Packet to Delivery Ratio (PDR)*

In this section we have presented the impact on packet to delivery ratio for LSLC protocol for three different scenarios that are as follows:

- Impact of data aggregation.
- Impact of number of levels.
- Impact of node density.

The Figure 4.6a reveals that the PDR of LSLC algorithm with data aggregation is approximately 23.07% lower than that of scenario when no data aggregation is used.



(a)

(b)

(c)

Figure 4.6: Impact on Packet to Delivery Ratio (PDR), (a) With data aggregation and without data aggregation (b) Number of levels (c) Node density

This is due to the fact that the data aggregation allows a lesser number of packets to be transmitted towards the base station. Although the packet generated for both the scenario is same but majority of packets are aggregated at different cluster head.

While on the other hand, the PDR of LSLC algorithm keeps on increasing as we increase the number of levels as shown in Figure 4.6b. The 10 level LSLC algorithm is almost having the

PDR of 37.5% higher than that of level 5. This is because with an increase in level the number of cluster heads increase and thereby the number of packets that flow in the network also increases.

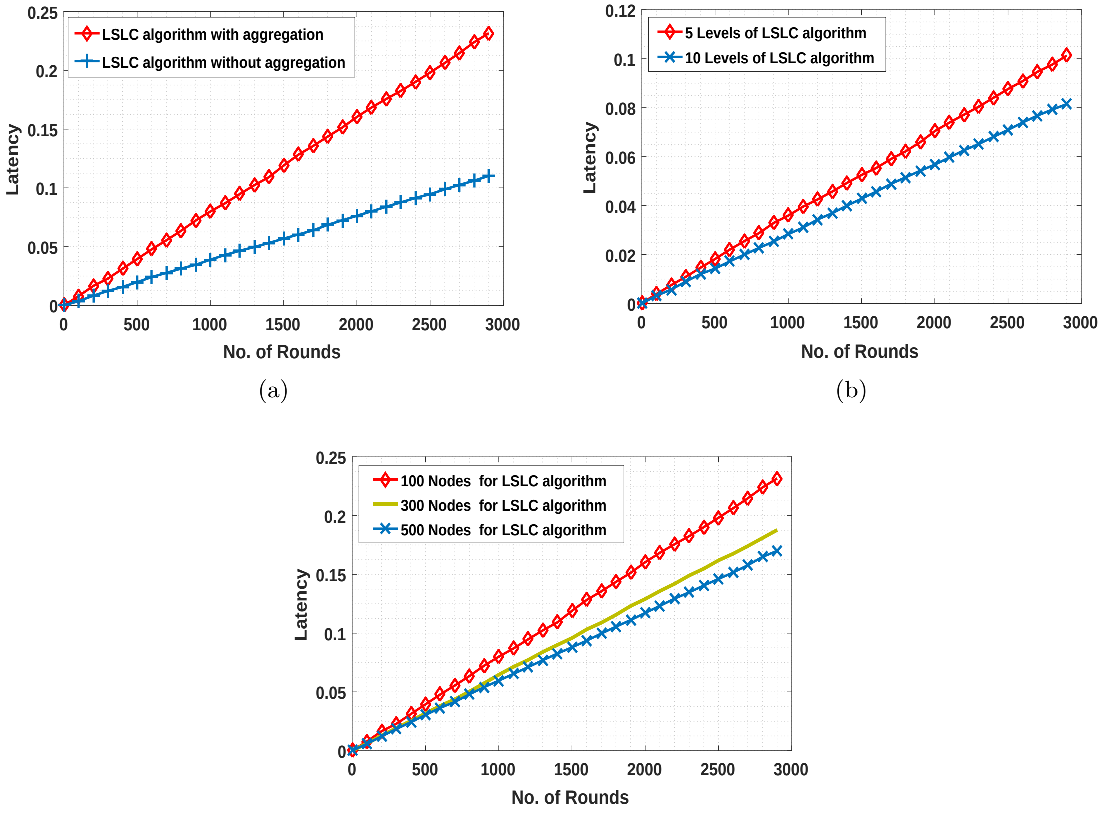
The Figure 4.6c shows the impact of packet to delivery ratio with an increase in node density as the number of nodes increases the packets generated by that nodes also increases. The results show that PDR improvement of 41.7% and 50% for 300 nodes, 500 nodes network respectively.

4.2.2.4 *Impact on Latency*

In this section we have presented the impact on latency for LSLC protocol for three different scenarios that is as follows:

- Impact of data aggregation.
- Impact of number of levels.
- Impact of node density.

The Figure 4.7a shows that there exists a tradeoff between the latency and the data aggregation.



(a)

(b)

(c)

Figure 4.7: Impact on Latency, (a) With data aggregation and without data aggregation (b) Number of levels (c) Node density

The Figure 4.7a reveals the fact that latency in the network almost get doubled for the scenario when data aggregation is performed in comparison to the case without data aggregation. This is due to the fact that the CH nodes take some time to apply aggregation function and the latency further increase if the CH nodes increases.

As the number of levels of LSLC is increased. It is analyzed from the Figure 4.7b that latency in the 10 levels LSLC algorithm is less in comparison to 5 levels. In round 2500, 10 levels LSLC algorithm has 23.52% less latency than 5 levels LSLC algorithm. Because in 10 levels LSLC algorithm more number of cluster heads are there which perform fast data aggregation as compared to 5 level LSLC algorithm due to which the latency is reduced and also the interlevel cluster head can choose the nearest CH easily because in 10 level LSLC more number of cluster head is there. This further reduces the interlevel communication delay.

As the number of nodes increases for LSLC algorithm the latency shows the downward trend as shown in Figure 4.7c. The latency is higher for 100 nodes network in comparison to 300 and 500 nodes network i.e. 20% for 300 nodes and 30% for 500 nodes w.r.t 100 nodes network . This is because more number of nodes are involved for a sensing task and thus the information pertaining to that specific task will reach the B.S with lesser time.

Summary of impact of results on LSLC algorithm The Table 4.3 summarizes the impact of simulation results on LSLC algorithm. In this table ↑, ↓ shows the increasing and decreasing trends of different performance metrics of LSLC algorithm.

Table 4.3: Summary of impact of results on LSLC algorithm

Impact on	**With and Without Data aggregation**		**Number of Levels**		**Node Density**	
	With Aggregation	Without Aggregation	5 Levels	10 Levels	100 Node	300,500 Node
Network Lifetime	↑	↓	–	–	–	–
Residual Energy	↓	↑	↓	↑	↓	↑
PDR	↓	↑	↓	↑	↓	↑
Latency	↑	↓	↑	↓	↑	↓

The table provides a clear and precise version of the analysis of the impact of different network parameters on the LSLC algorithm. The network lifetime provided by LSLC algorithm is better when data aggregation is considered. The LSLC algorithm with data aggregation is an energy efficient algorithm as the residual energy is less in comparison to the scenario of no data aggregation. Moreover, for the lesser number of levels as well as for lesser node density the residual energy is minimum. In addition to this, the PDR is less when data aggregation is performed and also its value remain lesser when the number of levels and node density are less.

In contrast to this, the latency of LSLC algorithm is higher when data aggregation is performed and this latency decreases as the number of levels and node density increases.

LSLC algorithm is an energy efficient algorithm for with data aggregation, during the increase in the number of levels to 5 and for 100 nodes. Because when the network dies then a very less energy is left. PDR is less for with data aggregation, for 5 levels, for 100 nodes. This is due to the reason of data aggregation is performed by CH and it is beneficial for bandwidth savings of network. The latency of LSLC algorithm shows a decreasing trend for all the three scenario that is no data aggregation, increase in the number of nodes as well as increase in the number of levels. Overall, we can say that LSLC algorithm with data aggregation for 100 node network and 5 levels provides considerable improvement in result for certain parameters expect latency.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This chapter summarizes the research of data aggregation that is carried out in this thesis report. The data aggregation is one of the method by which energy of the WSN node can be efficiently utilized. Moreover, in data aggregation, the selection of data aggregator node is performed on the basis of node degree and residual energy. Also, the aggregated data is beneficial for reducing the number of packets sent to B.S which further reduces the transmission energy of the data aggregator nodes. The aim of our research is to design centralized data aggregation algorithm that can enhance the network lifetime, reduce the network bandwidth as well as reduce the residual energy. We have done a broader review of the literature on various data aggregation algorithms and also we have summarized the same in the thesis. Moreover, we have proposed a centralized LSLC algorithm that attempts to efficiently utilize the available network energy before the lifetime of the network expires. The simulation results of LSLC algorithm is analyzed in comparison with existing algorithms. The results show that the LSLC provides a considerable improvement in terms of network lifetime, PDR, a packet sent to B.S and residual energy. Although the latency of LSLC is higher than that of other algorithms.

Moreover, we have also computed and analyzed the impact of different parameters like aggregation, node density, number of levels on the performing of LSLC protocol. The results show that the network lifetime provided by LSLC algorithm is better when data aggregation is considered. The LSLC algorithm with data aggregation is an energy efficient algorithm as the residual energy is less in comparison to the scenario of no data aggregation. Moreover, for the lesser number of levels as well as for lesser node density the residual energy is minimum. In addition to this, the PDR is less when data aggregation is performed and also its value remain lesser when the number of levels and node density are less. In contrast to this, the latency of LSLC algorithm is higher when data aggregation is performed and this latency decreases as the number of levels and node density increases.

LSLC algorithm is an energy efficient algorithm for with data aggregation, during the increase in the number of levels to 5 and for 100 nodes. Because when the network dies then a very less energy is left. PDR is less for with data aggregation, for 5 levels, for 100 nodes. This is due to the reason of data aggregation is performed by CH and it is beneficial for bandwidth savings of network. The latency of LSLC algorithm shows a decreasing trend for all the three scenario that is no data aggregation, increase in the number of nodes as well as increase in the number of levels. Overall, we can say that LSLC algorithm with data aggregation for 100 node network and 5 levels provides considerable improvement in result for certain parameters expect latency.

Furthermore, we would like to implement our algorithm for a heterogeneous network, the network with mobile nodes as well as multiple sinks.

References

[1] Anastasi G *et al.* (2009). Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks*, 7(3), 537-568.

[2] Ahmad, BA. Wireless sensor networks (WSN) and applications. Available at *http : //microcontrollerslab.com/wireless − sensor − networks − wsn − applications/* (Accesed on 19th September 2017).

[3] Williams, J. To wireless sensor networks and its applications. Available at *https : //wirelessmeshsensornetworks.wordpress.com/* (Accesed on 25th September 2017).

[4] Cui J. Data aggregation in wireless sensor networks. PhD Thesis, INSA Lyon, 2016.

[5] Xiao S, Li B and Yuan X (2015). Maximizing precision for energy-efficient data aggregation in wireless sensor networks with lossy links, *Ad Hoc Networks*, 26, 103-113.

[6] Tan H and Krpeolu I (2003). Power efficient data gathering and aggregation in wireless sensor networks, *ACM Sigmod Record*, 32(4), 66-71.

[7] Chao S *et al.* (2010). Energy efficient clustering algorithm for data aggregation in wireless sensor networks, *The Journal of China Universities of Posts and Telecommunications*, 17, 104-122.

[8] Madden S *et al.* (2002). TAG: A tiny aggregation service for ad-hoc sensor networks, *ACM SIGOPS Operating Systems Review*, 36(SI), 131-146.

[9] Mantri DS, Prasad NR and Prasad R (2016). Mobility and heterogeneity aware cluster-based data aggregation for wireless sensor network, *Wireless Personal Communications*, 86(2), 975-993.

[10] Ye M *et al.* (2005). EECS: an energy efficient clustering scheme in wireless sensor networks, *IEEE International Performance, Computing, and Communications Conference* [24th: Phoenix, AZ, USA: 2005],pp. 535-540.

[11] Arghavani M *et al.* (2017). Optimal energy aware clustering in circular wireless sensor networks, *Optimal energy aware clustering in circular wireless sensor networks*, 65, 91-98.

[12] Sasirekha S and Swamynathan S (2017). Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network, *Journal of Communications and Networks*, 19(4), 392-401.

[13] Randhawa S and Jain S (2017). DAHDA: Dynamic Adaptive Hierarchical Data Aggregation for Clustered Wireless Sensor Networks, *Wireless Personal Communications*, 97(4), 6369-6399.

[14] El Fissaoui M, Beni-Hssane A and Saadi M (2017). Mobile Agent Protocol based energy aware data Aggregation for wireless sensor networks, *Procedia Computer Science*, 113, 25-32.

[15] Gupta GP, Misra M and Garg K (2017). Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks, *Computers & Electrical Engineering*, 64, 262-276.

[16] Intanagonwiwat C *et al.* (2002). Impact of network density on data aggregation in wireless sensor networks, *International Conference on Distributed Computing Systems* [22^{nd}: Vienna, Austria: 2002], pp. 457-458.

[17] Saginbekov S and Jhumka A (2017). Many-to-many data aggregation scheduling in wireless sensor networks with two sinks, *Computer Networks*, 123, 184-199.

[18] Singh VK, Verma S and Kumar M (2016). Privacy preserving in-network aggregation in wireless sensor networks, *Procedia Computer Science*, 94, 216-223.

[19] Al-Karaki JN *et al.* (2004). Data aggregation in wireless sensor networks-exact and approximate algorithms, *High Performance Switching and Routing* [Phoenix, AZ, USA: 2004], pp. 241-245.

[20] Liao WH, Kao Y and Fan CM (2008). Data aggregation in wireless sensor networks using ant colony algorithm, *Journal of Network and Computer Applications*, 31(4), 387-401.

[21] Jian-ye Y *et al.* (2017). UKF sensor fusion method based on principal component analysis, *Communication and Information Processing* [3^{rd}: Tokyo, Japan: 2017], pp. 247-251.

[22] Li H, Lin K and Li K (2011). Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks, *Computer Communications*, 34(4), 591-597.

[23] Yousefi H *et al.* (2012). Structure-free real-time data aggregation in wireless sensor networks, *Computer Communications*, 35(9), 1132-1140.

[24] Zhang P *et al.* (2018). A secure data collection scheme based on compressive sensing in wireless sensor networks, *Ad Hoc Networks*, 70, 73-84.

[25] Wang P, He Y and Huang L (2013). Near optimal scheduling of data aggregation in wireless sensor networks, *Ad Hoc Networks*, 11(4), 1287-1296.

[26] Zhang J *et al.* (2017). A secret confusion based energy-saving and privacy-preserving data aggregation algorithm, *Chinese Journal of Electronics*, 26(4), 740746.

[27] Xu H *et al.* (2010). Energy-efficient cooperative data aggregation for wireless sensor networks, *Journal of Parallel and Distributed Computing*, 70(9), 953-961.

[28] Xiao S, Li B, and Yuan X (2015). Maximizing precision for energy-efficient data aggregation in wireless sensor networks with lossy links, *Ad Hoc Networks*, 26, 103113.

[29] Paczek B and Berna M (2014). Uncertainty-based information extraction in wireless sensor networks for control applications, *Ad Hoc Networks*, 14, 106-117.

[30] Heinzelman WR *et al.* (2000). Energy-efficient communication protocol for wireless microsensor networks. *Annual Hawaii International Conference on System Sciences* [33^{rd}: Maui, HI, USA: 2000] pp. 10-pp.

[31] Shah T *et al.* (2012). Energy efficient sleep awake aware (EESAA) intelligent sensor network routing protocol, *International Multitopic Conference* [15^{th}: Providence, Rhode Island, USA: 2012], pp. 317-322.

[32] Mahmood D *et al.* (2013). MODLEACH: a variant of LEACH for WSNs, *International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, [8^{th}: Compiegne, France: 2013], pp. 158-163.

[33] Hussain RH and Yousif AB (2015). Comparison the Performance of two Wireless Sensor Networks protocols (LEACH and EAMMH), *J.Thi-Qar Sci.*, 5(3), 98-108.

[34] Intanagonwiwat C *et al.* (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. *International Conference Mobile computing and networking* [6^{th}: Boston, MA, USA] pp. 56-67.

[35] Lindsey S *et al.* (2002). PEGASIS: Power-efficient gathering in sensor information systems, *IEEE Aerospace conference* [Big Sky, MT, USA: 2002], pp. 3-3.

[36] Selim SZ and Ismail MA (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1), 81-87.

[37] Peiravi A, Mashhadi HR and Javadi HS (2013). An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm, *International Journal of Communication Systems*, 26(1), 114-126.

[38] Misra R *et al.* (2006). Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks, *International Conference in Wireless and Optical Communications Networks* [Bangalore, India: 2006], pp. 5-pp.

[39] Intanagonwiwat C *et al.* (2003). Directed diffusion for wireless sensor networking, *IEEE/ACM Transactions on Networking (ToN)*, 11(1), 2-16.

[40] He W *et al.* (2007). Pda: Privacy-preserving data aggregation in wireless sensor networks, *International Conference on Computer Communications*, [26^{th}: Barcelona, Spain: 2007], pp. 2045-2053.

[41] Huang SH *et al.* (2007). Nearly constant approximation for data aggregation scheduling in wireless sensor networks, *IEEE international conference on computer communications* [26^{th}: Barcelona, Spain: 2007], pp. 366-372.

[42] He T *et al.* (2003). SPEED: A stateless protocol for real-time communication in sensor networks, *International Conference Distributed Computing Systems* [23^{rd}: Providence, Rhode Island, USA: 2003] pp. 46-55.

[43] G. Smaragdakis, I. Matta and A. Bestavros, “SEP: A stable election protocol for clustered heterogeneous wireless sensor networks”.

[44] Qing L, Zhu Q and Wang M (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks, *Computer communications*, 29(12), 2230-2237.

[45] Manjeshwar A *et al.* (2001). TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, *IEEE* [2001], p. 30189a

[46] Lu J *et al.* (2007). An improved immune-genetic algorithm for the traveling salesman problem, *IEEE international conference on Natural Computation* [3^{rd}: Haikou, China: 2007], pp. 297-301.

[47] Chen CS *et al.* (2010). Self-optimization in mobile cellular networks: Power control and user association, *IEEE international conference on Communications (ICC)* [Cape Town, South Africa: 2010], pp. 1-6.

[48] Qu W *et al.* (2007). An efficient method for improving data collection precision in lifetime-adaptive wireless sensor networks, *IEEE international conference on Communications* [Glasgow, UK: 2007], pp. 31613166.

[49] Mantri D *et al.* (2012). Two Tier Cluster based Data Aggregation (TTCDA) in wireless sensor network, *In Advanced Networks and Telecommuncations Systems (ANTS)* [Bangalore, India: 2012], pp. 117-122.

[50] Kumar D, Aseri TC, and Patel RB (2011). EECDA: energy efficient clustering and data aggregation protocol for heterogeneous wireless sensor networks, *International Journal of Computers Communications & Control*, 6(1), 113-124.

[51] Guo C *et al.* (2015). Compressive imaging with complex wavelet transform and turbo AMP reconstruction, *European Signal Processing Conference* [23^{rd}: Nice, France: 2015], pp. 1751-1755.

[52] Heinzelman WB, Chandrakasan AP and Balakrishnan H (2002). An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on wireless communications*, 1(4), 660-670.

[53] Wagstaff K *et al.* (2001). Constrained k-means clustering with background knowledge, *International Conference on Machine Learning* [8^{th}: Berkshires,Sandisfield, MA, USA: 2001], pp.577-584.

[54] Girao J *et al.* (2005). CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks. *IEEE International Conference on Communications* [Seoul, South Korea: 2005], pp. 3044-3049.

[55] Castelluccia C *et al.* (2005). Efficient aggregation of encrypted data in wireless sensor networks. 3rd Intl. *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Sensor Networks* [3^{rd}: Italy: 2005].

[56] Liu CX *et al.* (2013). High energy-efficient and privacy-preserving secure data aggregation for wireless sensor networks, *International Journal of Communication Systems*, 26(3), 380-394.

[57] Mykletun E *et al.* (2006). Public key based cryptoschemes for data concealment in wireless sensor networks, *International Conference on Communications* [Istanbul, Turkey: 2006], pp. 2288-2295.

[58] Roy S *et al.* (2014). Secure data aggregation in wireless sensor networks: Filtering out the attacker's impact, *IEEE Transactions on Information Forensics and Security*, 9(4), 681-694.

[59] Poornima AS *et al.* (2010). SEEDA: Secure end-to-end data aggregation in Wireless Sensor Networks, *International Conference on Wireless And Optical Communications Networks* [7^{th}: Colombo, Sri Lanka: 2010], pp. 1-5.

[60] Gilbert EN and Pollak HO (1968). Steiner minimal trees, *SIAM Journal on Applied Mathematics*, 16(1), 1-29.

[61] Yu B *et al.* (2009). Distributed data aggregation scheduling in wireless sensor networks. *IEEE INFOCOM* [Rio de Janeiro, Brazil: 2009], pp. 2159-2167.

[62] Wan PJ, Alzoubi KM and Frieder O. (2004). Distributed construction of connected dominating set in wireless ad hoc networks, *Mobile Networks and Applications*, 9(2), 141-149.

[63] Younis O and Fahmy S (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on mobile computing* 3(4), 366-379.

[64] Storn R and Price K (1997). Differential evolutiona simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization* 11(4), 341359.

[65] Smiti A *et al.* (2012). An improved clustering method based on gaussian means and dbscan techniques, *IEEE International Conference on Intelligent Engineering Systems (INES)* [16^{th}: Lisbon, Portugal: 2012], pp. 573-578.

[66] Sheikhpour R, Jabbehdari S and Khademzadeh A (2012). A Cluster-Chain based Routing Protocol for Balancing Energy Consumption in Wireless Sensor Networks, *International Journal of Multimedia and Ubiquitous Engineering* 7(2), 1-16.

[67] Jung T *et al.* (2013). Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation, *IEEE INFOCOM* [Turin, Italy: 2013], pp. 26342642.

[68] Upadhyayula S and Gupta SK (2007). Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks, *Ad Hoc Networks* 5(5), 626648.

[69] Qi H and Wang F (2001). Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks, *Proceedings of the IEEE* 18(5), 147-153.

[70] Chen M *et al.* (2007). Mobile agent-based directed diffusion in wireless sensor networks, *EURASIP Journal on Advances in Signal Processing* (1), 13p.

[71] Chen M *et al.* (2011). Itinerary planning for energy-efficient agent communications in wireless sensor networks, *IEEE Transactions on Vehicular Technology* 60(7), 32903299.

[72] Mpitziopoulos A *et al.* (2007). Deriving efficient mobile agent routes in wireless sensor networks with noid algorithm, *International Symposium on Personal, Indoor and Mobile Radio Communications* [18^{th}: Athens, Greece: 2007], pp. 1-5.

[73] Chen M *et al.* (2010). Directional source grouping for multi-agent itinerary planning in wireless sensor networks, *IInternational Conference on Information and Communication Technology Convergence (ICTC)* [Jeju, South Korea: 2010], pp. 207-212.

[74] Konstantopoulos C *et al.* (2010). Effective determination of mobile agent itineraries for data aggregation on sensor networks, *IEEE Transactions on Knowledge and Data Engineering* 22(12), 16791693.

[75] Gavalas D *et al.* (2016). Energy-efficient multiple itinerary planning for mobile agents-based data aggregation in wsns, *Telecommunication Systems* 63(4), 531545.

[76] Luo G *et al.* (2009). Compressive data gathering for large-scale wireless sensor networks, *International Conference on Mobile Computing and Networking* [15^{th}: Beijing, China: 2009], pp. 145156.

[77] Adewumi OG *et al.* (2013). RSSI based indoor and outdoor distance estimation for localization in WSN, *IEEE International Conference on In Industrial Technology (ICIT)* [Cape Town, South Africa: 2013], pp. 1534-1539.

[78]Boukerche A *et al.* (2007). Localization systems for wireless sensor networks. *IEEE wireless Communications*, 14(6).

[79] Tripathi RK. Base station positioning, nodes localization and clustering algorithms for wireless sensor networks. Ph.D. Thesis, IIT Kanpur, India, 2012.

[80] Spolsky J and Atwood J. Stack Exchange. Avaialble at $https://math.stackexchange.com/questions/543961/determine-third-point-of-triangle-when-two-points-and-all-sides-are-known$ (Accessed on 19^{th} September 2017).

[81] Mahajan S, Malhotra J and Sharma S. (2014). An energy balanced QoS based cluster head selection strategy for WSN, *Egyptian Informatics Journal*, 15(3), 189-199.

LIST OF PUBLICATIONS

SCI Journal

1. Mandeep Kaur, Amit Munjal, *"Data Aggregation Algorithms for Wireless Sensor Sensor Network: A Review"*, Ad Hoc Networks, Elsevier, [SCI, IF 3.047] [Communicated]
2. Mandeep Kaur, Amit Munjal, *"Levelling and Self Localization based Clustering algorithm (LSLC) algorithm for Data Aggregation in Wireless Sensor Network"*, Wireless Personal Communication, Springer, [SCI, IF 1.2] [Communicated]

Thesis

ORIGINALITY REPORT

15%	%	15%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Chao SHA, Ru-chuan WANG, Hai-ping HUANG, Li-juan SUN. "Energy efficient clustering algorithm for data aggregation in wireless sensor networks", The Journal of China Universities of Posts and Telecommunications, 2010 Publication	1%
2	Computer Communications and Networks, 2009. Publication	1%
3	Sukhchandan Randhawa, Sushma Jain. "Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status and Future Directions", Wireless Personal Communications, 2017 Publication	1%
4	Singh, Samayveer, Satish Chand, and Bijendra Kumar. "Energy Efficient Clustering Protocol Using Fuzzy Logic for Heterogeneous WSNs", Wireless Personal Communications, 2015. Publication	1%