Performance Analysis of Periodic Sensor Networks Using Different Data Aggregation Techniques

[1]Sangeetha J, [2]Mrs Pushpalatha S,
[1]Mtech Student, [2]Assistant Professor,
[1]Digital Electronics and Communication, Master of Technology,
[1]VTU Post Graduation Center, Mysuru, India

Abstract: Wireless sensor network is almost everywhere, these networks has huge application in habitat monitoring, disaster management, security and military, etc. Data aggregation is gaining much attention from researchers as best way to reduce the huge volume of data generated in WSNs by eliminating the redundancy among sensing data. In this paper, we suggest an efficient data aggregation technique for clustering-based periodic wireless sensor networks. Further in a local aggregation at sensor node level, our technique allows cluster head to eliminate redundant data sets generated by neighboring nodes by applying two data aggregation methods. These suggested methods are based on the sets similarity functions and the distance functions, respectively. Based on real sensor data, we have analyzed their performances according to the energy consumption, delay, accuracy and overhead, and we show how these methods can significantly improve the performance of sensor networks.

***IndexTerms*- Periodic Sensor Network (PSN), data aggregation, clustering topology, similarity and distance functions**

I. INTRODUCTION

Wireless sensor networks (WSNs) has been widely used in a various number of applications such as healthcare, target tracking, environment monitoring, military surveillance, industrial and agricultural producing and intelligent home furnishing, etc. Such networks generally consist of a large number of wireless sensor nodes and a few sink nodes. Clustering is now considered as an efficient topology control method in WSN that has more advantages, especially as far as scalability and network maintenance are concerned, compared to other topologies. However, most of the present data aggregation techniques are based on clustering topology are dedicated to event driven data model and the main focus is on the selection of CHs. In these techniques only CHs process and aggregate data without any processing at the level of the nodes themselves [1].

In this paper, we consider a cluster-based periodic sensor network (CPSN), in which each sensor monitors the given phenomenon and periodically sends its collected data to its CH. Then, we introduce two layer algorithms one at the node level and another at the CH level. In the first level, an aggregation process aggregates data on a periodic basis avoiding each sensor node to send its raw data to the sink. In the second level, we present and compare two different methods to search for redundancies between data sets given by neighbouring sensor nodes.

II. LITERATURE SURVEY

This section describes the previous work done on data aggregation techniques.
Mingxin Yang uses data fusion algorithm FTDA, this algorithm uses cluster based topology. It helps to extend the network lifetime and save energy consumption. Algorithm uses exponential smoothing prediction model which plays a predictive role in the data transmission phase. Nodes in the network have same energy and nodes in the network use clock synchronization to transfer data [2].

Taochun Wang, Ji Zhang, Yonglong Luo, Kaizhong Zuo, Xintao Ding uses secure and itinerary based data aggregation algorithm. The algorithm mainly based on dynamic network topology. It gives low communication overhead and energy consumption, yet high safety and accuracy. Here last node encrypts the aggregated result, and then returns the cipher text back to the sink [3].

Sercan Vancin, Ebubekir Erdem uses SEED algorithm. Algorithm is based on cluster based topology. It increases the network efficiency. Here only one sensor node awakes and relays obtained data to the cluster head and the remaining sensor nodes stays in sleepy mode. Here the network configuration is the rotating head is chooses on the basis of residual energy of the sensor node [4].

Dragos I. Sacaleanu, Dragos M. Ofrim, Rodica Stoian, Vasile Lazarescu uses adaptive routing algorithm (ARA). It uses grid network based topology. The algorithm takes in to account the residual energy of the sensor nodes. Here the network configuration is the random deployed sensors grouped in grid layouts [5].

Ali Norouzi, Faezeh Sadat Babamir, Zeynep Orman uses genetic algorithm. It uses tree based static topology. Here algorithm is made up of fitness and selection function. The network configuration consists of chromosome which is the collection of nodes and each node has a parent id [6].

Selvakumar Sasirekha and Sankaranarayanan Swamynathan uses cluster-chain mobile agent routing algorithm. Algorithm makes use of the advantages of both LEACH and PEGASIS. Here the WSN divides in to few clusters and runs in two phases. Here the performance metrics such as energy consumption, transmission delay and network lifetime can be improved [7].

III. PROPOSED SCHEME

Reducing energy consumption is an important issue in WSNs where sensors are mainly battery-limited. Hence, data aggregation is an important operation in WSNs used to decrease the data transmission, thus, to increase the network lifetime. It means computing and transmitting partially aggregated data to the end user rather than transmitting raw data in networks to reduce the energy consumption. Data aggregation in wireless sensor networks has been well studied in recent years. The performance of any data aggregation technique is mainly dependent on the network's topology. Here we are using cluster topology and two data

aggregation techniques. Aggregation takes place in two level one is sensor node level and CH level. The aggregation techniques used here are advanced k –means and advanced Euclidean distance based data aggregation techniques.

The project is divided in to five main modules as shown in fig 1:

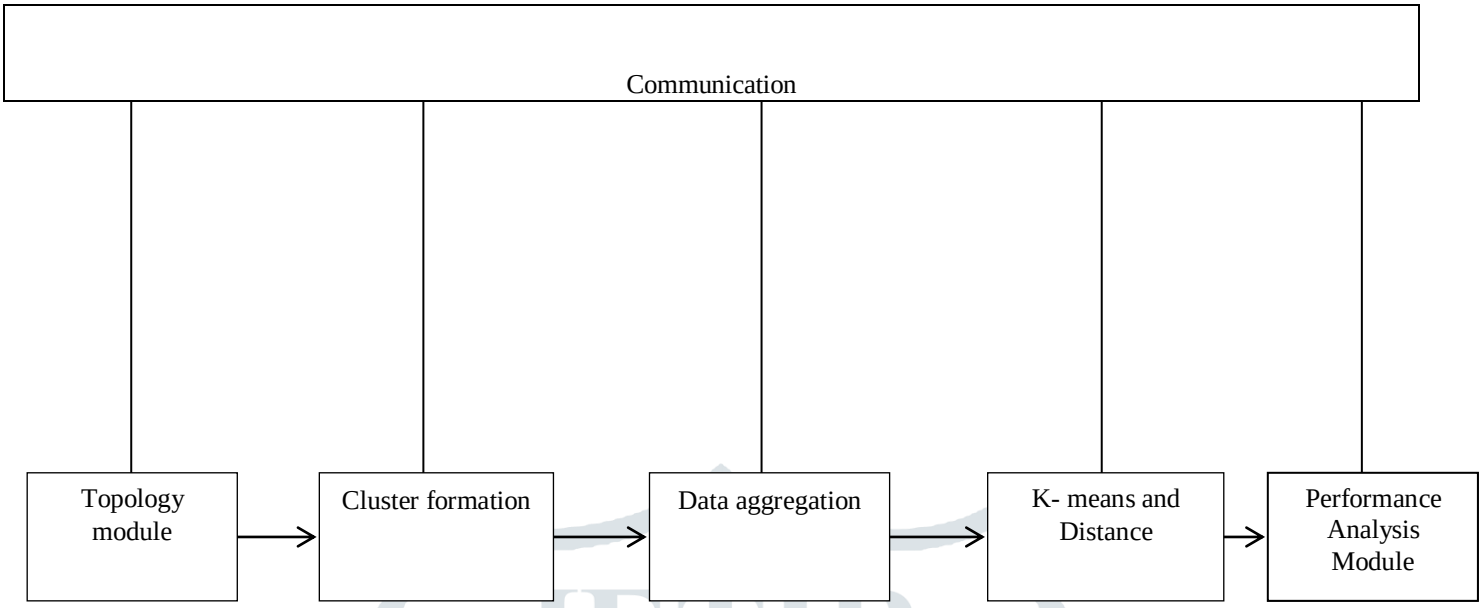

Fig 1: Project module

3.1 Cluster Formation Algorithm

The cluster-based architecture is used to construct the topology. Nodes cooperate to form clusters, and each cluster consists of a CH along with some Cluster Members (CMs) located within the transmission range of their CH. Before nodes can join the network, they have to acquire valid certificates from the CA, which is responsible for distributing and managing certificates of all nodes, so that nodes can communicate with each other unrestrainedly in a MANET. In this model, if a node proclaims itself as a CH, it propagates a CH Hello Packet (CHP) to notify neighboring nodes periodically. The nodes that are in this CH's transmission range can accept the packet to participate in this cluster as cluster members. On the other hand, when a node is deemed to be a CM, it has to wait for CHP. Upon receiving CHP, the CM replies with a CM Hello Packet (CMP) to set up connection with the CH. Afterward, the CM will join this cluster; meanwhile, CH and CM keep in touch with each other by sending CHP and CMP in the time period Tu.

In this algorithm, the network nodes are partitioned according to an artificial hierarchical structure with the sole purpose of reducing the number of entries in the routing tables. The entire network is divided into partitions like regions and districts. Cluster Formation algorithm divides the entire are into multiple zones. Each Cluster having a set of nodes in its zone. Cluster based algorithm which is responsible for deploying the nodes. The entire area is divided into zones with each zone bounded with the limits with some xmin and xmax. The y region is bounded within the limits ymin and ymax. Each Cluster is allocated a set of nodes.
The Cluster Formation algorithm follows the following steps:

1. Number of Clusters, vertical and horizontal end points for each of the cluster acts as an input.
2. For each of the Cluster Formation Step3, 4 and 5 are repeated until all clusters are formed
3. Pick appropriate x end points and y end points for the cluster.
4. Generate the cluster id for the cluster.
5. Execute the Node Deployment algorithm and place the nodes within the cluster.

3.2 Node Distribution Algorithm

The Node Distribution algorithm is used to linearly disperse the nodes across the network.

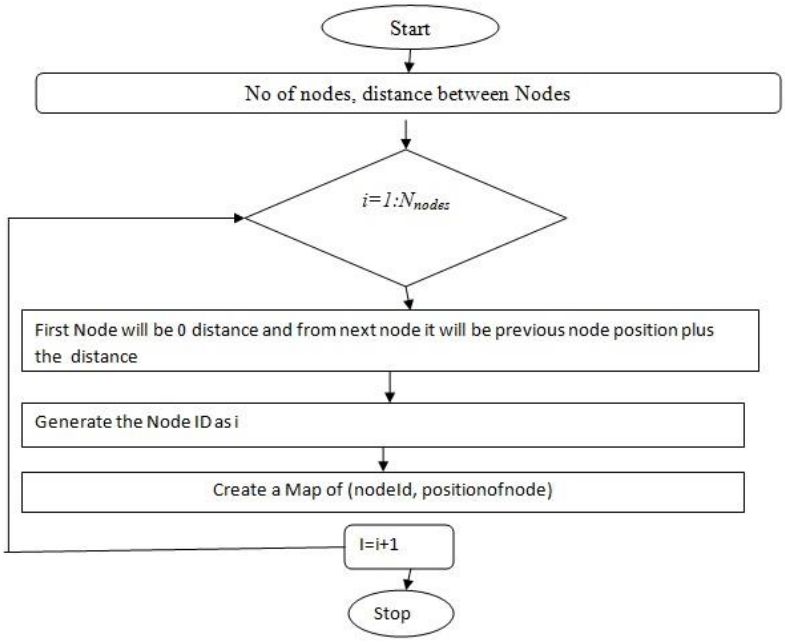

Fig 2: Flow chart for nodes Deployment Algorithm

The pseudo code for initialization of nodes randomly is as follows

a. for i=0 to num_of_nodes
b. Initialize mobile nodes
c. Enable random motion
d. Nodes Energy.
e. Define initial node position
f. Initialize agent
g. Attach agent to node.
h. End for.

3.3 Energy Module

Energy Module, as implemented in, is a node attribute. The energy module represents level of energy in a mobile host. The energy module in a node has an initial value which is the level of energy the node has at the beginning of the simulation. This is known as initial Energy. It also has a given energy usage for every packet it transmits and receives. These are called transmission power and receive power. The energy module only maintains the total energy and does not maintain radio states. It is generic enough for future simulations such as the CPU power consumption. Please note that the old energy module indeed maintains some radio states, and have some methods to manipulate them, and they are only used by the adaptive fidelity module. This approach may cause inconsistency with wireless-phy. To keep adaptive fidelity work, we did not remove it from the energy module, but it is obsolete, and should not be used further. Now all access to the energy module should go through wireless-phy.

3.4 Software and hardware requirements

3.4.1 Software Requirements

- Operating System : UBUNTU 11.10
- Tool: NS-2.35
- Language : TCL and AWK

An OTcl script will do the following.

- Initiates an event scheduler.
- Sets up the network topology using the network objects.
- Tells traffic sources when to start/stop transmitting packets through the event scheduler.

The basic function of AWK is to search files for lines (or other units of text) that contain certain patterns. When a line matches one of the patterns, awk performs specified actions on that line. AWK keeps processing input lines in this way until the end of the input files are reached. Programs in awk are different from programs in most other languages, because awk programs are data-driven; that is, you describe the data you wish to work with, and then what to do when you find it. Most other languages are procedural; you have to describe, in great detail, every step the program is to take. When working with procedural languages, it is

usually much harder to clearly describe the data your program will process. For this reason, awk programs are often refreshingly easy to both write and read. When you run awk, you specify an awk program that tells awk what to do. The program consists of a series of rules. Syntactically, a rule consists of a pattern followed by an action. The action is enclosed in curly braces to separate it from the pattern. Rules are usually separated by newlines. Therefore, an awk program looks like this:

pattern { action }
pattern { action } ...

3.4.2 Hardware requirements

- Processor: Any processor with speed above 500 MHz.
- RAM: 512Mb.
- Hard Disk: 20 GB.
- Input device: Standard Keyboard and Mouse.
- Output device: VGA and High Resolution Monitor.

3.5 Data aggregation techniques

Aggregation takes place in two levels: one is at sensor level and another at CH level

3.5.1 Aggregation at sensor level

Data transmission between the member nodes and their CH or between CH and sink is mainly based on single hop communication. Member nodes will collect the data in a periodic manner. Each member node will sends its data to the CH at each period. Sensor node will sense the environment at a fixed sampling rate.

For each new captured measurement, a sensor node searches the similarities of the new taken measurement. If a similar measurement is found, it deletes the new one and increments the corresponding weight by 1. After local aggregation sensor will transform the initial vector of collected data Ai=[ai1,ai2,ai33,ai4,ai5] in to final aggregated data in sensor node Ai`= {(ai1`,wgt(ai1`), ai2`,wgt(ai2`), …….aik`,wgt(aik`)}. The weight of a measurement aij is defined as the number of similar measures to aij in the same vector Ai.

3.5.2 Aggregation at CH level-advanced k means algorithm

(1) Arbitrarily generate points (cluster centers), being the number of clusters desired.
(2) Calculate the distance between each of the data points to each of the centers, and assign each point to the closest center.
(3) Calculate the new cluster center by calculating the mean value of all data points in the respective cluster.
(4) With the new centers, repeat step (2). If the assignment of cluster for the data points changes, repeat step (3); else stop the process.

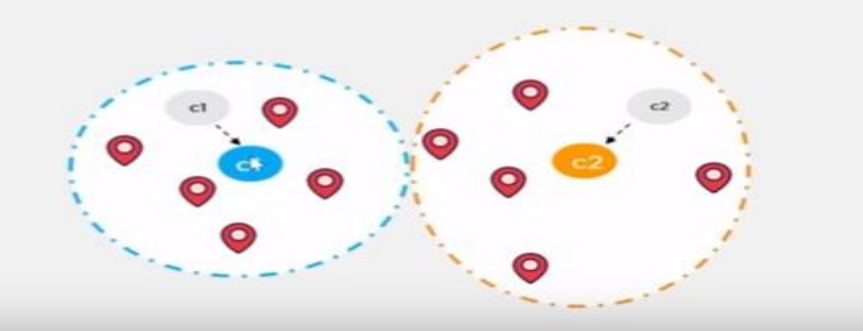

Fig 3: K-means algorithm topology

3.5.3 Advanced Euclidean distance based algorithm

The Euclidean distance is the ordinary distance that is the straight line distance, between two points, sets or objects. It is used in applications and domains like computer vision and prevention of identity theft. The Euclidean distance is already used in WSN during the deployment phase in terms of sensors localization and inter-sensors distance estimations .CH considers that two sets are redundant if the normalized distance between them is less than the threshold. Every distance function has its own method to calculate the distance between data sets. Straight-line distance in Euclidean distance and the angle between data sets in Cosine distance. CH chooses that one having the highest cardinality to send it to the sink. CH assigns to each set its weight when sending it to the sink. Then it adds it to the list of sets to be sent to the sink. After that, it removes all pairs of redundant sets. At last the CH assigns to each set its weight when sending it to the sink.

IV. RESULTS AND DISCUSSION

Parameters	Advanced k-means algorithm	Advanced Euclidian distance algorithm
Throughput	290.46	596.00
Delay	1357ms	53.316ms
PDF	52%	96%
Overhead	27.091	7.647
Energy	37.161	19.392

Table 1: Parameter values obtained by two different data aggregation algorithms

NNode

Neighbor Detail

Source	Neighbor	SX-Pos	SY-Pos	H-Distance(d)
0	1	101	307	70
0	2	101	307	62
0	3	101	307	68
0	4	101	307	117
0	5	101	307	127
0	7	101	307	282
0	8	101	307	246
0	9	101	307	223
0	10	101	307	211
0	12	101	307	212
0	13	101	307	296
0	16	101	307	254
0	17	101	307	158
0	19	101	307	260
0	20	101	307	191
0	21	101	307	258
0	49	101	307	235
0	50	101	307	295
0	51	101	307	252
0	52	101	307	236
0	53	101	307	158
0	54	101	307	165
0	55	101	307	207
0	56	101	307	118
0	57	101	307	193
0	58	101	307	148
0	59	101	307	229
0	60	101	307	127
1	0	140	366	70
1	2	140	366	74
1	3	140	366	119
1	4	140	366	139

Plain Text

Table 2: Table for distance between source and neighbor

Graphs for different parameters:

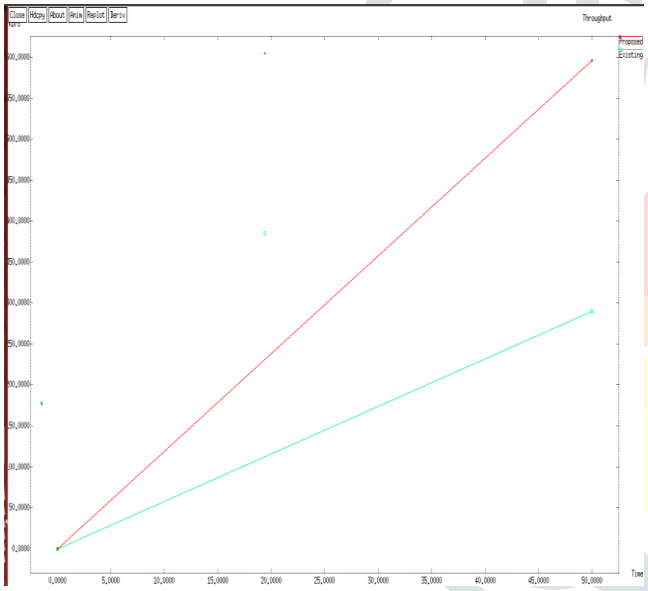

Fig 4a: Throughput graph

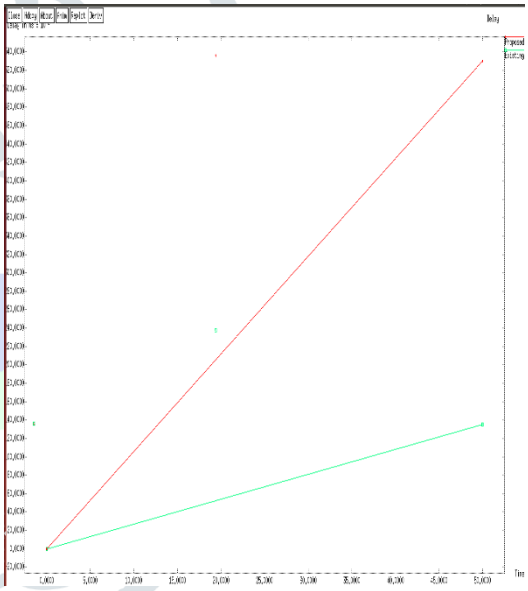

Fig 4b: Delay graph

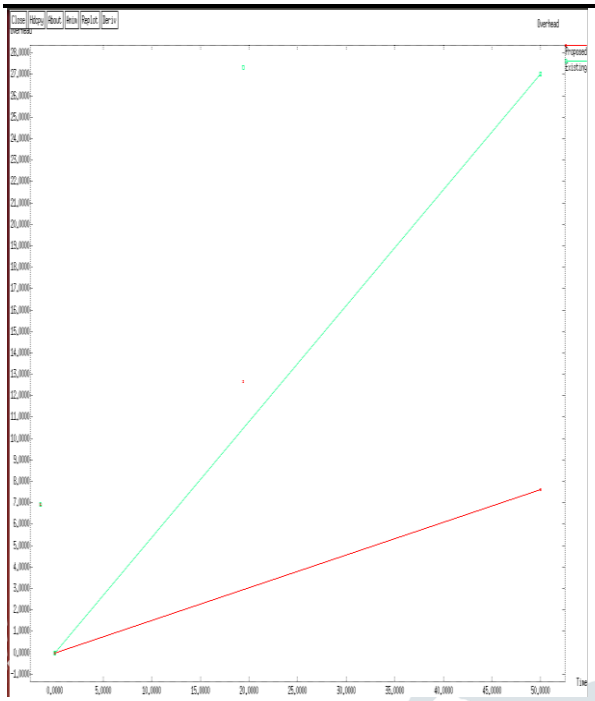

Fig 4c: overhead graph

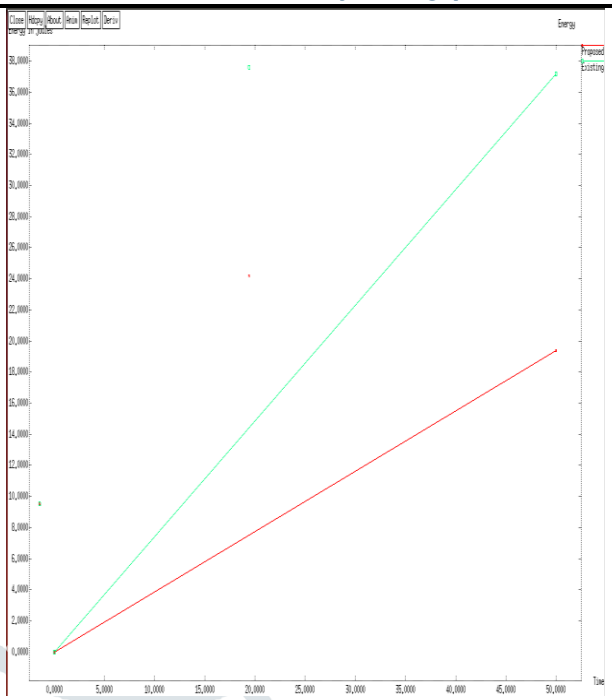

Fig 4d: Energy graph

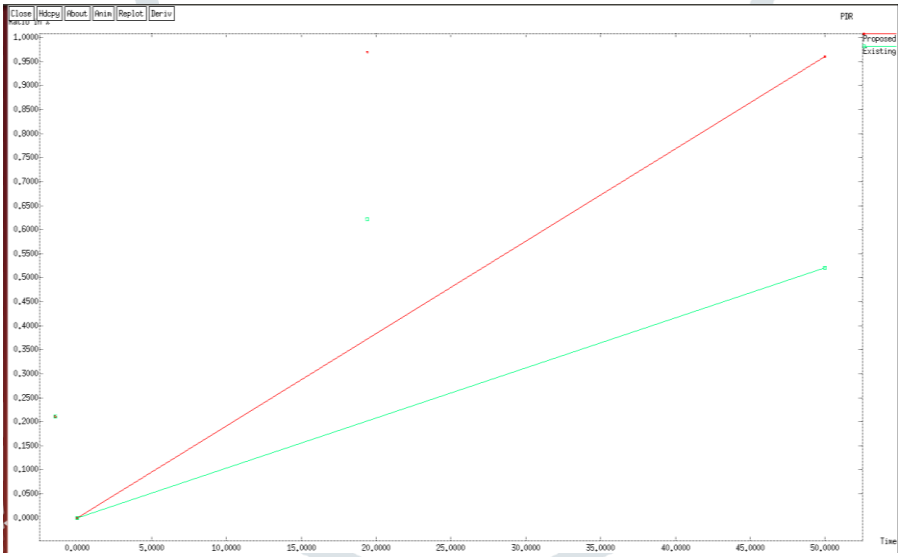

Fig 4e: PDF graph

V. CONCLUSION

Data aggregation is very essential for WSNs where huge amount of data collected by the sensors need to be minimized. After eliminating redundant data collected by sensor we propose two data aggregation technique allowing cluster head to eliminate the redundant data. The proposed methods based on the similarity function and distance function. In future we can make the nodes generating data will not be active at the same time. Sensor nodes conserve more energy compared to other methods.

VI. ACKNOWLEDGMENT

This project is partly supported by the assistant professors Department of Electronics and communication Systems of Visvesvaraya Technological University, Post Graduation Center, Sathagalli Mysuru.

REFERENCES

[1]. Hassan Harb, Abdallah Makhoul, Samar Tawbi, and Raphael Couturier, “Comparison of Different Data Aggregation Techniques in Distributed Sensor Networks”, IEEE, 2017.
[2]. Mingxin Yang, “Data Aggregation Algorithm for Wireless Sensor Network Based on Time Prediction”, IEEE, 2017.

[3]. Taochun Wang, Ji Zhang, Yonglong Luo, Kaizhong Zuo, Xintao Ding, "An Efficient and Secure Itinerary-based Data Aggregation Algorithm for WSNs", IEEE, 2017.
[4]. Sercan Vancin Ebubekir Erdem, "Performance Analysis of the Energy Efficient Clustering Models in Wireless Sensor Networks", IEEE, 2017.
[5]. Dragos I. Sacaleanu, Dragos M. Ofrim, Rodica Stoian, Vasile Lazarescu, "Increasing lifetime in grid wireless sensor networks through routing algorithm and data aggregation techniques", International Journal Of Communications, 2011.
[6]. Ali Norouzi, Faezeh Sadat Babamir, Zeynep Orman, "A Tree Based Data Aggregation Scheme for Wireless Sensor Networks Using GA", scientific research, 2012.
[7]. Selvakumar Sasirekha and Sankaranarayanan Swamynathan,"Cluster-Chain Mobile Agent Routing Algorithm for Efficient Data Aggregation in Wireless Sensor Network", Journal of Communications and Networks, august 2017.