

DISTRIBUTED LOAD BALANCING ALGORITHM FOR WIRELESS SENSOR NETWORK

K. Navaz¹, S. Athinarayanan², S. Sameena³ and R. Kavitha⁴

^{1,2}Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Tirupati, India

³Department of Computer Science and Engineering, Universal College of Engineering and Technology, Vallioor, India

⁴Department of Computer Science, The Madurai Diraviyam Thayumanavar Hindu College, India

Abstract

A Wireless Sensor Network (WSN) comprises of spatially scattered autonomous sensors to screen physical or natural conditions and to amiably go their information through the system to a Base Station. Grouping is a basic assignment in Wireless Sensor Networks for vitality effectiveness and system quality. Grouping through Central Processing Unit in remote sensor systems is outstanding and being used for quite a while. In this paper, we propose a few procedures that balance the vitality utilization of these hubs and guarantee greatest system lifetime by adjusting the activity stack as similarly as could be expected under the circumstances. Directly grouping through dispersed strategies is being produced for conveying with the issues like system lifetime and vitality. In our work, we connected both concentrated and conveyed *k*-means clustering calculation in system test system. *K*-means is a model based algorithm that surrogates between two noteworthy advances, passing on perceptions to groups and processing cluster focuses until the point when a ceasing standard is satisfied. Improved results are accomplished and related which demonstrate that conveyed clustering is compelling than brought together grouping.

Keywords:

Wireless Sensor Network, Clustering, K-Means, Network Stability

1. INTRODUCTION

Wireless Sensor Network (WSN) contains of two classes of hubs, to be specific essential and optional hubs. Essential hubs very much delegated with sensor and radio framework. The Secondary hubs are just the sending hubs which have a radio alone to go about as discontinuous (connect) hubs. These hubs made animated the rise of Wireless Sensor Network (WSNs) in applications including ecological checking, war zone examination, atomic, natural and synthetic assault identification, human services and home applications. WSN is made with the controls out of restricted vitality [1], memory [1], preparing power [2], and data transmission for correspondence [2], and radio range [2]. As sensors must work under strict power requirements, transmitting data detected to end station may be in feasible. This inspires to search for creating resources by utilizing grouping calculations sharing data in single-bounce neighbours only. Clustering is the blend of comparable articles and a grouping of a set is a parcel of its components that is chosen to limit some proportion of variety [3]. Clustering calculations are frequently helpful in applications in different fields, for example, computerized reasoning, perception, learning hypothesis, PC illustrations, neural systems, design acknowledgment and measurements. Commonsense applications [12] of grouping incorporate example order under unsubstantiated learning, quickness seek, time arrangement investigation, content mining and heading finding.

Clustering in sensor hubs has been broadly chased by the exploration network with the end goal to understand the adaptability, vitality and lifetime issues of sensor systems. Clustering calculations limit the correspondence in a nearby area and transmit just fundamental data to whatever is left of the system through the sending hubs. A gathering of hubs frames a group and the nearby co operations between cluster individuals are controlled through a cluster head (CH). Cluster [4] individuals by and large speak with the group head and the gathered information are amassed and joined by the cluster go to monitor vitality. The cluster heads can likewise shape another layer of groups among themselves before achieving the sink. The Fig.1 shows the architecture of sensed data forwarding with clustering and aggregation.

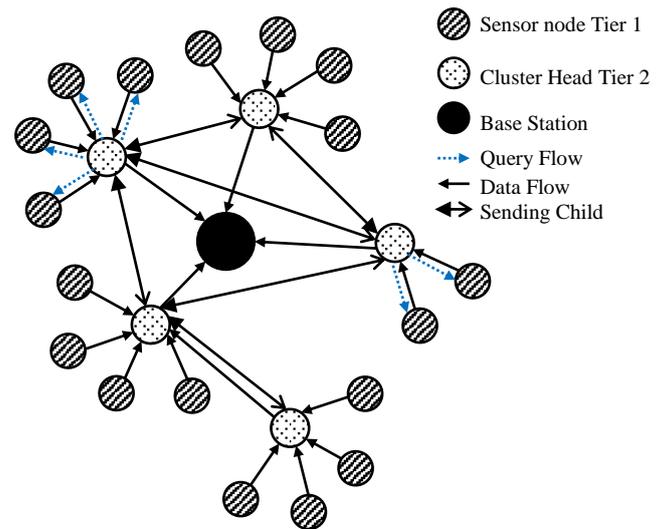


Fig.1. Sensed Data forwarding with clustering and aggregation

Issue is set on partitional grouping algorithms, which yield a single parceling of the information characterized by a settled number of parameters [4] [13]. With these parameters being not exactly the accessible information, partitional grouping can bear the cost of promising conveyed utilization of deterministic methodology. A prevalent incorporated also appropriated deterministic partitional clustering approach is offered by the *k*-means algorithm, which highlights straightforward, profoundly dependable, and quick focalized repetitions and re-grouping amid disappointment states.

The rest of this paper is as follows. In section 2, we evaluate related works and their highlights offered to clustering methods and the section 3 manages the concentrated method for grouping hubs utilizing *k*-means calculation. In section 4, the computational model of the *k*-means algorithm is presented. Section 5 exhibits

the relative outcomes and examination of these outcomes created in like manner to disseminated k -means clustering algorithm and centralized approach. In section 6, we concluded with conclusive remarks.

2. RELATED WORKS

Clustering is done to relate comparative hubs and spares vital vitality squandered in direct information transmission to the base station. Nodes in the system set up themselves into progressive level structures. Inside a specific group, information conglomeration and sending are performed at cluster go to decrease the measure of information transmitting to the base station. Group development is normally founded on outstanding vitality of sensor hubs and sensors nearness to cluster head [6]. Hubs other than cluster head pick their group head directly after organization and transmit detected data to the cluster head. The job of group head, acting naturally a sensor hub, is to forward this information and its very own information to the base station in the wake of performing information total and sending.

An Energy Efficient Scheduling for Cluster-Tree WSN is proposed in [7]. Clustering in this technique utilizes Cluster-tree development. Cyclic Scheduling for information transmission in Zigbee condition utilizing Time-division Multiple Access (TDMA). The cluster is active only once during the schedule period leads to so called cyclic behaviour of periodic schedule when there are the streams with inverse heading in a WSN. Adaptive behaviour of the planning issue when new assignments are added to the first timetable and the portability of sensor hub or the switch is not addressed. A two level design connect with dynamic nature correspondence [8] addresses the blame tolerant target following. For Clustering LEACH based scheme issued to organize the nodes. SNs may flop because of vitality exhaustion equipment disappointment, correspondence interface blunders, and malevolent assaults. A runtime recuperation component is proposed, which distinguishes blames in entryways and recoups sensors from fizzled groups by relegating them to sound passages without re-clustering the framework. The most of the time utilized blame tolerant system for WSN is the organization of excess/surplus SNs. At the point when repetitive hubs (RNs) are given, at that point the BS can acquire information; regardless of whether a few SNs are flopped because of any reason. Message overhead isn't tended to in this iterative strategy for grouping. Portability Based Structure with group choice [9] is done dependent on Mobility and Residual Energy. Cluster clusters are made by considering (i) Link Stability and (ii) Connection time – by investigating the Packet Loss. MBC is a subsidiary of Cluster Based Routing Protocol (CBR). The Proposed calculation gives the disseminated preparing prompts choice of two group heads in a similar territory if their individual parameters are same. In secure information gathering, versatile information authority is utilized to gather the information from the non-group head hubs. A common key utilized between the hubs. Tree based sensor key administration procedure is utilized [10] proposes grouping plans have Time stamp convention (TSP), polynomial focuses sharing convention (PPSP) and mystery sharing convention (SSP). Expanded complexity in calculation presentation and Energy productivity is low. Multilayer clustering presents parcel of hub organization in a similar zone of intrigue. Pretty much looks like the more established adaptation of utilizing portals between

clusters [11] have addressed the Hotspot problem effectively in WSN Clustering.

There are huge numbers of sensors deploy in network which sense the neighboring information and processed the gathered data to the base station, processing speed, storage and communication bandwidth. Because of these constraints fast transmission, fault tolerant, load balancing mechanism and energy efficient routing are critical requirements in WSNs to increase the network lifetime. By considering this limitation, it is necessary to design novel energy aware and fault tolerant routing protocol for heterogeneous sensor network. In [16] a survey on various existing techniques are studied and the research challenges are listed out.

In [15], the author proposed a clustering technique which balances the load among the cluster by using some backup nodes. The backup high energy and high processing power nodes replace the cluster head after the cluster reaches to its threshold limit. This approach increases the network lifetime and provides high throughput. In [14], wireless sensor networks still many issues are there. In this survey Load balancing issue is considered. Load balancing typically is not equally distributing load on all nodes in the network, it involves identifying the optimal load on nodes of the network to increase its lifetime. This paper describes about load balancing techniques such as cluster based, protocol based and algorithm based.

3. CLUSTERING ALGORITHMS

The k -means algorithm is based mainly on the Euclidian distances and cluster head selection depends on residual energies of nodes [12]. So here the central node collects the information about the node id, position and remaining energy of all nodes and stores this information in a list in the central node. After getting this information from all nodes it starts performing the clustering algorithm (k -means) [13].

- Step 1:** If we want to cluster the nodes into ' k ' clusters, take ' k ' number of centroids initially at random places
- Step 2:** Calculate the Euclidian distance from each node to all centroids and assign it to centroid nearest to it. By this ' k ' initial clusters are formed. Suppose there are n nodes are given such that each one of them belongs to R_d . The problem of finding the minimum variance clustering of this nodes into k clusters is that of finding the k centroids $\{m_j\}_{k_j=1}^k$ in R_d such that, where $d(X_i, m_j)$ denotes the Euclidean distance between X_i and m_j . The points $\{j\}_{k_i=1}^k$ are known as cluster centroids or as cluster means.
- Step 3:** Recalculate the positions of centroids in each cluster and check for the change in position from the previous one
- Step 4:** If there is change in position of any centroid then go to Step 2, else the clusters are finalized and the clustering process ends. By this the clustering of nodes into k number of clusters is done [13] and the cluster heads in each cluster are to be chosen as shown in Table.2.

3.1. CENTRALIZED K-MEANS CLUSTERING

When a centralized authority makes choices and partitions the nodes into clusters without the participation of other nodes is

centralized way of clustering. Here the centralized authority gets the necessary information for clustering from the individual nodes. Based on this information it will cluster by some algorithm and sends the clustering results back to the individual nodes.

- *Cluster Head Selection*: From the nodes which are at the first distance level and the next distance level from the centroid, we take the highest energy nodes and elect the one which is nearer as the cluster head.
- *Declaration of Cluster head*: After the central node completes the process of clustering and selecting cluster head, the central nodes ends back the information under which cluster it belongs and its cluster head to each node individually. Thus every node knows under which cluster it belongs and its cluster head and this completes the process of clustering in a centralized way.

3.2. DISTRIBUTED K-MEANS CLUSTERING

Distributed clustering is the mechanism in which, there is no fixed central CH and the CH keeps on changing from node to node based on some pre-assigned parameters. In this section, literature survey of various published distributed clustering algorithms for WSNs is presented, based on some advantages like efficient utilization of communication bandwidth within the clusters, avoiding redundant message transfer between the sensor nodes, localizing energy efficient route setup within the clusters, reduction in energy consumption etc.

When every node participates in making clustering decisions, it is distributed way of clustering. Here every node gets the necessary information for clustering from all other nodes. Based on this information all nodes will cluster by some algorithm and also decides the cluster head. Since the k -means algorithm [13] is based on Euclidian distances and energies (for choosing cluster head), the information about the positions and energies of all nodes is obtained by every node by exchanging messages among themselves. After getting the information about all nodes every node runs the algorithm (k -means). The k -means algorithm for clustering and the algorithm for choosing cluster head are similar to the algorithms used in centralized clustering. As every node runs the same algorithm, every node knows under which cluster it belongs and its cluster head. So here there is no process of sending back as in centralized. Thus the distributed clustering process is complete.

4. SIMULATION ENVIRONMENT

We simulate the proposed algorithm using NS2. We develop source code to implement the centralized and distributed k -means clustering as follows. Steps for implementing centralized and distributed clustering is as follows

Step 1: *Sending the position and energy of each node to central node:*

The decision making location have position and energy of each node in the network. In the centralized clustering the central node acts as the decision making ability. So the position and energy of all nodes should be available to that central node. In this we made the

positions and energy of all nodes available to central node by the following steps.

In the distributed clustering all nodes participate in the decision making process. So that every node has the position and energy of all other nodes. In this work, we made the positions and energy of all nodes available to every node by using following steps.

- a. *Accessing positions and energy*: In simulator, we can declare any number of nodes and we can place where ever we want. We declare nodes, place them and create a scenario to perform the clustering of these nodes. The position of each node can be accessed from its own object files created. Here we can access the positions by calling some predefined functions in the mobile node.h and energy from energy model.h.
- b. *Place to store those values*: For the central node/individual nodes (in distributed k -mean) to store the values of node id, position and energy values of all nodes, we created a structure in form of linked list in node.h and its initialized pointer is declared in class Node. So whenever we want to record data (node id, position, energy), we allocate the space dynamically and store those values into it.
- c. *Forming Packet*: For the sending the information by node about its node id, position and energy, it needs to access that information, form packet using the collected data and then send it.
- d. *Sending through Routing*: For the nodes to transmit and receive data, we need to initialize agent and attach it to the node. For the transmitted data to reach the destination (central node), we need to attach both the agents of source node and destination node before sending. The packet follows the predefined routing protocol (Modified AODV in our case) to reach destination.
- e. *Updating the List*: When the transmitted packet is received at the destination node (i.e. central node /individual nodes), it accesses the content of the packet. The packet is checked for redundancy and updated to the list created if new.

Step 2: *Cluster Head Selection:*

After the centroid positions are finalized in the clustering process, we consider nodes which are at the nearest distance and also the next nearest distance from the centroid. The node with highest energy is considered as Cluster Head. If more than one node in the two levels has the highest energy, then the node nearest to the centroid is selected as cluster head. If more than one node has the highest energy in the same level of distance, then the node with the least node id is selected as cluster head.

Step 3: *Declaration of Cluster Head:*

In centralized clustering, after the node completes the process of clustering and selecting cluster head, each node should get the information under which cluster it belongs and its cluster head. This information is given to each node by central node by repeating the process of attaching agent to sender and receiver, connecting them and sending. By this each node knows under which

cluster it belongs and its cluster head. This ends the process of clustering in centralized fashion.

In distributed clustering, after the centroid positions are finalized in the clustering process, we consider nodes which are at the nearest distance and also the next nearest distance from the centroid. The node with highest energy is considered as Cluster Head. If more than one node in the two levels has the highest energy, then the node nearest to the centroid is selected as cluster head. If more than one node has the highest energy in the same level of distance, then the node with the least node id is selected as cluster head. By this each node knows under which cluster it belongs and its cluster head. This ends the process of clustering in distributed fashion.

Table.1. Node Configuration Parameters

Parameter	Value
Topology	670×670m ²
k	3
Centroid 1	200,100,0 (x_1, y_1, z_1)
Centroid 2	90,500,0 (x_2, y_2, z_2)
Centroid 3	110,10,0 (x_3, y_3, z_3)
Routing	AODV (Modified)
Propagation	Two Ray Ground Model
Initial energy	10J
rxPower	0.3J
txPower	0.9J

5. PERFORMANCE EVALUATION

Time taken depends on number of nodes, positions of nodes, position of central node, and the position of initial centroids placed (in centralized). Since the set-up is same for both centralized and distributed clustering, the positions of nodes and initial centroids remain constant. Therefore, time taken is independent of positions of nodes. So the time taken to cluster just by changing the number of nodes can be measured. In Distributed clustering time taken includes time taken for exchanging control messages (i.e. time taken for exchanging the position and energy details with all nodes) and clustering time (i.e. time taken for computing algorithm), Here time value is measured from trace file by taking the average of time taken by two highest and two lowest time taking nodes multiplied by the total number of nodes.

In centralized clustering time taken includes sending time (i.e. time taken for sending positions and energies of all nodes to central node), clustering time (i.e. time taken for computing algorithm) and resending (i.e. time taken for the central node to send back the information of clustering to individual nodes). Here the sending time is measured by taking average of maximum and minimum values of time taken for sending, multiplied by the total number of nodes. Resending time is also measured by taking average of maximum and minimum values of time taken for sending from central node, multiplied by the total number of nodes, as shown in the Table.2. Time taken to cluster by varying the number of nodes show that the time taken for centralized clustering is more than the time for distributed clustering. This may be due to the resending of clustering information from central

node which is not needed in distributed, as all nodes do the clustering process individually.

Table.2. Time taken for Centralized and Distributed Clustering processes for varying number of nodes

No. of Nodes	Time Taken in Seconds	
	Distributed Clustering	Centralized Clustering
25	1.5	1.8
30	1.7	2.2
35	2.1	3.5
40	3.1	4
45	3.8	5.9
50	4.2	6.1
55	5	7.2
60	5.8	9
65	6	9.9

Average energy consumed depends on number of nodes, positions of nodes, position of initial centroids placed and also the position of central node (in centralized). Since the scenario is same for both centralized and distributed clustering, the positions of nodes and initial centroids remain constant. Therefore, energy consumed is independent of positions of nodes. So the average energy consumed per node to cluster just by varying the number of nodes can be measured. In general, for clustering, energy is consumed mainly for transmitting, receiving packets and also for processing as shown in the Table.3.

Here average energy consumed per node is measured by taking the difference between the total initial energies of all nodes and total final energies left in the nodes after clustering and dividing by total number of nodes.

Table.3. Average energy consumed per node of centralized and distributed clustering for varying number of nodes

No. of Nodes	Average Energy Consumption	
	Distributed Clustering	Centralized Clustering
25	0.4	0.3
30	0.5	0.5
35	0.64	0.6
40	0.79	0.8
45	0.88	0.9
50	1	1.2
55	1.38	1.4
60	1.58	1.6
65	1.79	1.8

Average energy consumed per node by varying the number of nodes shows that there is not much difference in the consumed energy for centralized and distributed clustering. This may be because the energy consumed in distributed clustering for exchanging of control messages (containing position and energy details) among all the nodes is almost equal to the energy consumed in both sending (each node) to central node and resending (from central node) to all nodes as shown in Table.4.

Table.4. Time taken for distributed and centralized clustering for varying number of clusters (k)

No. of Clusters (k)	Time taken for clustering in Seconds	
	Distributed Clustering	Centralized Clustering
2	5.5	8.9
3	5.6	8.9
4	5.5	8.9
5	5.6	8.8
6	5.5	8.9
7	5.5	8.9
8	5.5	8.9
9	5.5	8.9

Considering the time taken to cluster with varying k value shows that the time taken to cluster is same for all the k values (i.e. number of clusters) as shown in Table.4. This is because the processing time is negligible for both centralized and distributed k -means algorithm.

6. CONCLUSIONS

The network is more stable for distributed clustering when compared to centralized clustering. In centralized clustering if the central node malfunctions or dies then the entire network will fail whereas in distributed clustering failure of any node does not affect the entire network. In the centralized way of clustering if a packet drops while sending the node information to the central node or while resending back from central node to the individual nodes (i.e.it is more dependent on the routing algorithms), then the node will be left out. Where as in distributed clustering while exchanging the control messages the routing algorithms are not involved since when a node broadcasts its information, all the nodes which are in its receiving range will receive it and again broadcasts it. In this way the message travels the whole network. According to the three factors on result analysis, we concluded that distributed clustering consume less time and energy for its operation compared with centralized schema. In future some novel techniques to be adopted with distributed scheduling algorithm for further minimizing the delay.

REFERENCES

- [1] F. Zhao and L.J. Guibas, “*Wireless Sensor Networks: An Information Processing Approach*”, Morgan Kaufmann Publishers, 2004.
- [2] V. Raghunathan, C. Schurgers, S. Park and M.B. Srivastava, “Energy-Aware Wireless Micro Sensor Networks”, *IEEE Signal Processing Magazine*, Vol. 19, No. 2, pp. 40-50, 2002.
- [3] Jennifer Yick, Biswanath Mukherjee and Dipak Ghosal, “Wireless Sensor Network Survey”, *Computer Networks*, Vol. 52, No. 12, pp. 2292-2330, 2008.
- [4] A. Ameer Ahmed Abbasi and Mohamed Younis, “A Survey on Clustering Algorithms for Wireless Sensor Networks”, *Computer Communications*, Vol. 30, No. 14-15, pp. 2826-2841, 2007.
- [5] S. Bandyopadhyay and E.J. Coyle, “An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks”, *Proceedings of IEEE 22nd Annual Joint Conference on IEEE Computer and Communications Societies*, pp. 1713-1723, 2003.
- [6] O. Younis, M. Krunz and S. Ramasubramanian, “Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges”, *IEEE Networks*, Vol. 20, No. 3, pp. 20-25, 2006.
- [7] S.P. Lloyd, “Least-Squares Quantization in PCM”, *IEEE Transactions on Information Theory*, Vol. 28, No. 2, pp. 129-137, 1982.
- [8] C.R. Lin and M. Gerla, “Adaptive Clustering for Mobile Wireless Networks”, *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, pp. 1265-1275, 1997.
- [9] I.S. Dhillon and D.S. Modha, “A Data-Clustering Algorithm on Distributed Memory Multiprocessors”, *Large-Scale Parallel Data Mining*, pp. 245-260, 2000.
- [10] R.O. Duda, P.E. Hart and D.G. Stork, “*Pattern Classification*”, 2nd Edition, Wiley, 2002.
- [11] P.A. Forero, A. Cano and G.B. Giannakis, “Distributed Feature-based Modulation Classification using Wireless Sensor Networks”, *Proceedings of IEEE International Conference on Military Communications*, pp. 1-5, 2008.
- [12] P.A. Forero, A. Cano and G.B. Giannakis, “Consensus-Based-Means Algorithm for Distributed Learning using Wireless Sensor Networks”, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 11-14, 2008.
- [13] P.A. Forero, A. Cano and G.B. Giannakis “Consensus-based Distributed Expectation-Maximization Algorithm for Density Estimation and Classification using Wireless Sensor Networks”, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1989-1992, 2008.
- [14] C. Godsil and G. Royle, “*Algebraic Graph Theory*”, Springer, 2000.
- [15] Seema Shivapur, Suvarna G. Kanakaraddi and A.K. Chikaraddi, “Load Balancing Techniques in Wireless Sensor Networks: A Comparative Study”, *International Journal of Emerging Technology in Computer Science and Electronics*, Vol. 14, No. 2, pp. 21-28, 2015.
- [16] Dipak Wajgi and Nileshsingh V. Thakur, “Load Balancing based Approach to Improve Lifetime of Wireless Sensor Network”, *International Journal of Wireless and Mobile Networks*, Vol. 4, No. 4, pp. 12-17, 2012.
- [17] Roshani Talmale, M. Nirupama Bhat and Nita Thakare, “Analysis of Load Balancing and Fault Tolerant Routing Protocol for Wireless Sensor Network”, *Helix*, Vol. 8, No. 5, pp. 3946-3949, 2018.