k-MEANS CLUSTERING IN WIRELESS SENSOR NETWORKS

P. Sasikumar
School of Electronics Engineering
VIT University
Vellore, India
sasikumar.p@vit.ac.in

Sibaram Khara
School of Electronics Engineering
VIT University
Vellore, India
sibaramk@gmail.com

***Abstract*—A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions and to cooperatively pass their data through the network to a Base Station. Clustering is a critical task in Wireless Sensor Networks for energy efficiency and network stability. Clustering through Central Processing Unit in wireless sensor networks is well known and in use for a long time. Presently clustering through distributed methods is being developed for dealing with the issues like network lifetime and energy. In our work, we implemented both centralized and distributed k-means clustering algorithm in network simulator. k-means is a prototype based algorithm that alternates between two major steps, assigning observations to clusters and computing cluster centers until a stopping criterion is satisfied. Simulation results are obtained and compared which show that distributed clustering is efficient than centralized clustering.**

Keywords- wireless sensor network; clustering; ns-2; k-means; network stability

I. INTRODUCTION

Wireless sensor network (WSN) comprises of two classes of nodes, namely primary and secondary nodes. Primary nodes equipped with sensor and radio system. The Secondary nodes are simply the forwarding nodes which have a radio alone to act as intermittent (bridge) nodes. These nodes made prompted the emergence of wireless sensor networks (WSNs) in applications including environmental monitoring, battlefield surveillance, nuclear, biological and chemical attack detection, health care and home applications. WSN is poised with the constraints of limited energy [1], memory [1], processing power [2], and bandwidth for communication [2], and radio range [2]. Literature abounds in these constrain-based research issues. As sensors must operate under stringent power constraints, transmitting information sensed to end station may be infeasible. This motivates to search for creating resources by using clustering algorithms sharing information in single-hop neighbors only.

Clustering is the grouping of similar objects and a clustering of a set is a partition of its elements that is chosen to minimize some measure of dissimilarity [3]. Clustering algorithms are often useful in applications in various fields such as visualization, pattern recognition, learning theory, computer graphics, neural networks, artificial intelligence, and statistics. Practical applications [12] of clustering include pattern classification under unsupervised learning, proximity search, time series analysis, text mining and navigation.

Clustering in sensor nodes has been widely pursued by the research community in order to solve the scalability, energy and lifetime issues of sensor networks. Clustering algorithms limit the communication in a local domain and transmit only necessary information to the rest of the network through the forwarding nodes (gateway nodes). A group of nodes form a cluster and the local interactions between cluster members are controlled through a cluster head (CH) (a chosen leader). Cluster [4] members generally communicate with the cluster head and the collected data are aggregated and fused by the cluster head to conserve energy. The cluster heads can also form another layer of clusters among themselves before reaching the sink.

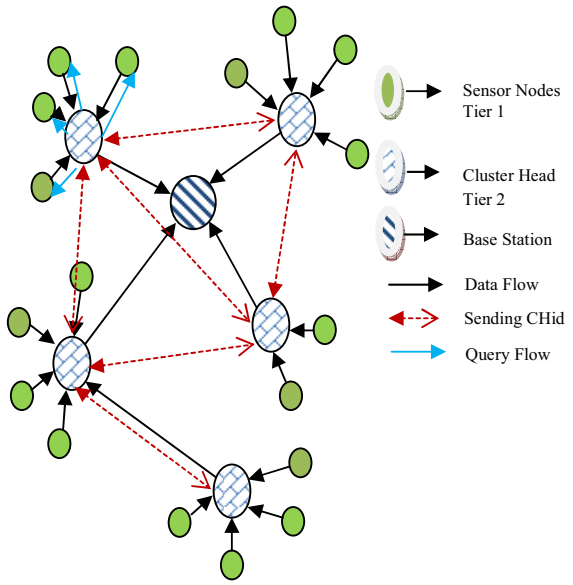

Figure 1. Sensed Data forwarding with clustering and aggregation

Issue is placed on partitional clustering algorithms (as opposed to regular hierarchical clustering), which yield a single partitioning of the data described by a fixed number of parameters [4][13]. With these parameters being less than the available data, partitional clustering can afford promising distributed implementation of deterministic approach. A popular centralized as well distributed deterministic partitional clustering approach is offered by the k-means algorithm, which features simple, highly reliable, and fast-convergent iterations & re-clustering during failure states [5].

978-0-7695-4850-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CICN.2012.136

The remainder of this paper is organized as follows. Section 2 analyzes related works and their features offered to clustering techniques. Section 3 deals with the centralized way of clustering nodes using k-means algorithm. In Section 4 we carefully analyze the computational complexity of the k-means algorithm. In other words, we show that as the number of data points increases the communication costs incurred by our parallelization strategy are relatively insignificant compared to the overall computational complexity with the distributed way of clustering nodes using k-means algorithm. Section 5 presents the comparative results and analysis of these results generated in common to distributed k-means clustering algorithm and centralized approach. Finally we have concluded with final comments in Section 6

II. RELATED WORK

Clustering is done to relate similar nodes and saves necessary energy wasted in direct data transmission to the base station. Nodes in the network organize themselves into hierarchical tier - structures. Within a particular cluster, data aggregation and forwarding are performed at cluster-head to reduce the amount of data transmitting to the base station. Cluster formation is usually based on remaining energy of sensor nodes and sensor's proximity to cluster-head [6]. Nodes other than cluster-head choose their cluster-head right after deployment and transmit sensed information to the cluster-head. The role of cluster-head, being itself a sensor node, is to forward these data and its own data to the base station after performing data aggregation and forwarding.

An Energy Efficient Scheduling for Cluster-Tree WSN is proposed in [7]. Clustering in this method uses Cluster-tree formation. Cyclic Scheduling for data transmission in Zigbee environment using Time-division Multiple Access (TDMA). The cluster is active only once during the schedule period leads to so called cyclic behavior of periodic schedule when there are the flows with opposite direction in a WSN. Adaptive behavior of the scheduling problem when new tasks are added to the original schedule and the mobility of sensor node or the router is not addressed

A two tier-architecture network with dynamic nature communication [8] addresses the fault tolerant target tracking. For Clustering LEACH based scheme is used to organize the nodes. SNs may fail as a result of energy depletion hardware failure, communication link errors, and malicious attacks. A runtime recovery mechanism is proposed, which detects faults in gateways and recovers sensors from failed clusters by assigning them to healthy gateways without re-clustering the system. The most frequently used fault-tolerant technique for WSN is the deployment of redundant/surplus SNs. When redundant nodes (RNs) are provided, then the BS is able to obtain data; even if some SNs are failed due to any reason. Message overhead is not addressed in this iterative method of clustering.

Mobility Based Structure with cluster selection [9] is done based on Mobility and Residual Energy. Cluster groups are created by considering (i) Link Stability and (ii) Connection time – by analyzing the Packet Loss. MBC is a derivative of Cluster Based Routing Protocol (CBR). The Proposed algorithm provides the distributed processing leads to selection of two cluster heads in the same area if their individual parameters are same.

In secure data collection, mobile data collector is used to collect the data from the non-cluster head nodes. A shared key used between the nodes. Tree based sensor key management technique is used. [10] proposes clustering schemes have Time stamp protocol (TSP), polynomial points sharing protocol (PPSP) and secret sharing protocol (SSP). Increased complexity in algorithm introduction and Energy efficiency is very low.

Multilayer clustering introduces lot of node deployment in the same area of interest. More or less resembles the older version of using gateways between clusters. [11] have addressed the Hot spot problem effectively in WSN Clustering.

III. K-MEANS ALGORITHM

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 residual 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-mean) [13].

Algorithm:

1. If we want to cluster the nodes into 'k' clusters, take 'k' number of centroids initially at random places
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 $\{ m_j \}^k{}_{j=1}$ in R_d such that,
   $$\left(\frac{1}{n}\right)\times\Sigma\left(\min_j d^2\left(X_i, m_j\right)\right), \text{ for i = } 1 \text{ to } n,$$
   where $d(X_i , m_j)$ denotes the Euclidean distance between X_i and m_j. The points $\{j\}^k{}_{i=1}$ are known as cluster centroids or as cluster means.
3. Recalculate the positions of centroids in each cluster and check for the change in position from the previous one
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 Fig. 2.

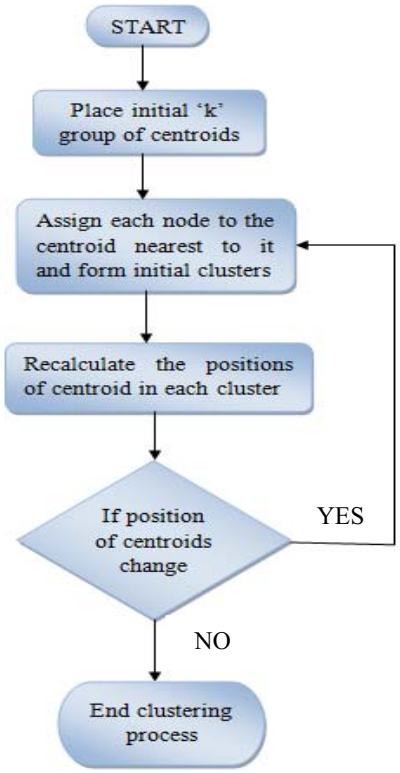

Figure 2. Flow chart showing sequence of k-means algorithm

IV. CENTRALIZED K-MEANS CLUSTERING

When a centralized authority makes decisions and partitions the nodes into clusters without the involvement 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 node sends 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.

V. DISTRIBUTED K-MEANS CLUSTERING

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-mean 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-mean).

The k-mean 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.

VI. SIMULATION SETUP

We simulate the proposed algorithm using ns-2 [14]. We develop source code to implement the centralized and distributed clustering as follows.

A. Steps for implementing centralized and distributed clustering

1. *Sending the position and energy of each node to central node:* The position and energy of each node should be available at the decision making location. In the centralized clustering the Sink node (central node) acts as the decision making authority. So the position and energy of all nodes should be available to the central node. In our work we made the positions and energy of all nodes available to central node by the following sequence of steps.

In the distributed clustering all nodes participates in the decision making process. So the position and energy of all nodes should be available to every node. In our work we made the positions and energy of all nodes available to every node by the following sequence of steps.

a) Accessing positions and energy: In network simulator, we can declare 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 accessed the positions by calling some predefined functions in the 'mobilenode.h' and energy from 'energymodel.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 node to send the information 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.

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.

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 I. NODE CONFIGURATION PARAMETERS

Parameter	Value
Topology	670x670 m2
k	3
Centroid 1	200,100,0 (x1,y1,z1)
Centroid 2	90,500,0 (x2,y2,z2)
Centroid 3	110,10,0 (x3,y3,z3)
Routing	AODV (Modified)
Propagation	TwoRayGround
Initial energy	10 J
rxPower	0.3 J
txPower	0.9 J

B. Assumptions made

The time taken for performing k-mean algorithm and for choosing cluster head is zero since the processing time is negligible. The time taken and average energy consumed is independent of the position of central node

VII. PERFORMANCE EVALUATION

Time taken 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 time taken is independent of positions of nodes. So the time taken to cluster just by varying 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.

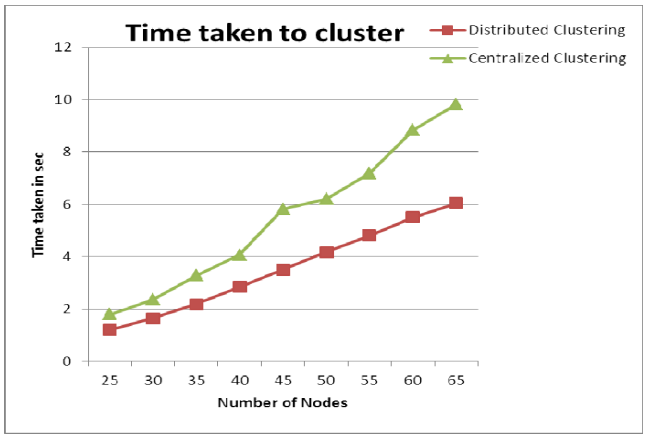

Figure 3. Time taken for Centralized and Distributed Clustering processes for varying number of nodes

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 Fig. 3.

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.

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 Fig.4.

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.

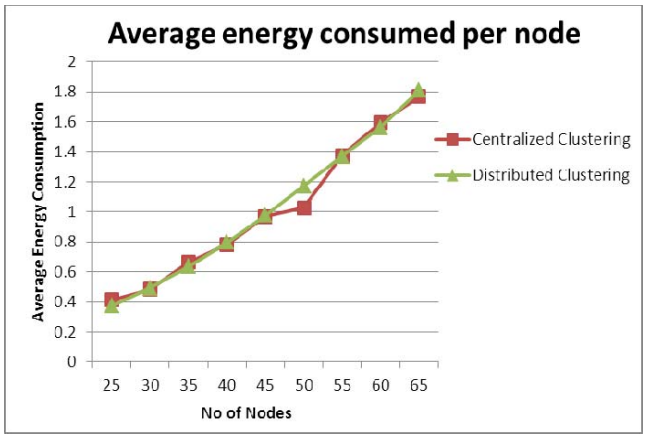

Figure 4. Average energy consumed per node of centralized and distributed clustering for varying number of nodes

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 Fig. 4.

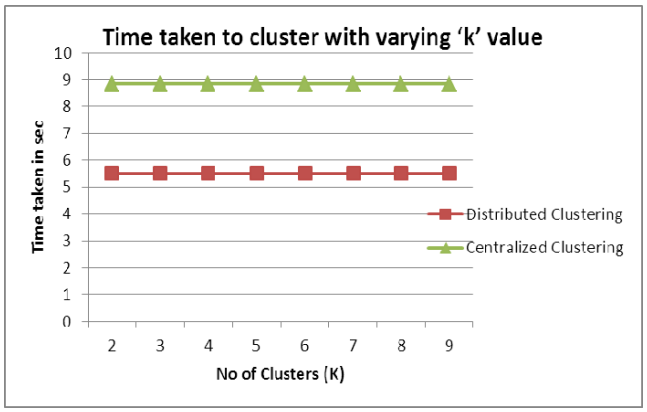

Figure 5. Time taken for distributed and centralized clustering for varying number of clusters, k

Considering 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 Fig.5. This is because the processing time is negligible for both centralized and distributed k-means algorithm.

VIII. Conclusion

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.

References

[1] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, "Wireless sensor network survey", Published by Elsevier, 14 April 2008, Page No: 2292 – 2330.

[2] A. Ameer Ahmed Abbasi, Mohamed Younis, "A survey on clustering algorithms for wireless sensor networks", Computer Communications- Published by Elsevier, 2007, Page No: 2826 – 2841

[3] S. Bandyopadhyay, E.J. Coyle,"An energy efficient hierarchical clustering algorithm for wireless sensor networks", IEEE INFOCOM, Volume 3, 2003, Pages 1713-1723

[4] O. Younis, M. Krunz, S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges", IEEE Network, Volume 20, Issue 3, May 2006, Pages 20-25

[5] S. P. Lloyd, "Least-squares quantization in PCM," IEEE Trans. Inf. Theory, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[6] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE Journal on Selected Areas in Communications 15 (7) (1997) 1265–1275.

[7] Zdenek Hanzalek, and Petr Jurcik, "Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee", IEEE TRANSACTIONS on industrial informatics, vol. 6, no. 3, august 2010

[8] S. Bhatti1 J. Xu, and M. Memon, "Clustering and fault tolerance for target tracking using wireless sensor networks", IET Wirel. Sens. Syst.,2011,Vol.1,Iss.2,pp.66–73

[9] S. Deng, J. Li, and L. Shen, "Mobility-based clustering protocol for wireless sensor networks with mobile nodes", IET Wirel. Sens. Syst., 2011, Vol. 1, Iss. 1, pp. 39–47

[10] A.S. Poornima and B.B. Amberker, " Secure data collection using mobile data collector in clustered wireless sensor networks", IET Wirel. Sens. Syst., 2011, Vol. 1, Iss. 2, pp. 85–95

[11] Y. Liu, N. Xiong, Y. Zhao, A.V. Vasilakos, J. Gao, and Y. Jia, "Multi-layer clustering routing algorithm for wireless vehicular sensor networks", IET Commun., 2010, Vol. 4, Iss. 7, pp. 810–816

[12] Manasi N. Joshi, "Parallel K - Means Algorithm on Distributed Memory Multiprocessors", Spring 2003

[13] Pedro A. Forero, Alfonso Cano, and Georgios B. Giannakis, "Distributed Clustering Using Wireless Sensor Networks", IEEE Journal Of Selected Topics In Signal Processing, Vol. 5, No. 4, August 2011

[14] http://www.isi.edu/nsnam/ns/