

# A Comparative Analysis of Hierarchical Agglomerative Clustering and Distributed HAC

<sup>1</sup>S.Aravindhan, <sup>2</sup>Dr.D.Maruthanayagam

<sup>1</sup>Research Scholar, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India

<sup>2</sup>Head/Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India

## ABSTRACT

In the hierarchical wireless sensor network (WSN), cluster-based network architecture can enhance network self-control capability and resource efficiency, and prolong the whole network lifetime. Thus, clustering has also been a topic of interest in many different disciplines. Finding an energy-effective and efficient way to generate cluster is very important in WSN. This paper describes the well-understood Hierarchical Agglomerative Clustering (HAC) algorithm by provide a Distributed HAC (DHAC) algorithm. With simple six-step clustering, DHAC provides a bottom-up clustering approach by grouping similar nodes together before the cluster head (CH) is selected. DHAC can accommodate both quantitative and qualitative information types in clustering, while offering flexible combinations using four commonly used HAC algorithm methods, SLINK, CLINK, UPGMA, and WPGMA. With automatic CH rotation and re-scheduling, DHAC avoids re-clustering and achieves uniform energy dissipation through the whole network.

**Keywords :** Wireless Sensor Network, Distributed Hierarchical Agglomerative Clustering (DHAC), Cluster Head, SLINK, CLINK, UPGMA and WPGMA.

## I. INTRODUCTION

Wireless Sensor Network (WSN) is a collection of micro sensor nodes for an intelligent autonomous monitoring system with communication and computing capabilities that are deployed to monitor the areas. It holds the potential to revolutionize many segments of our economy and life, from environmental monitoring. WSN is one of the most important information access platforms, and it is always deployed in extreme environment where people could not survive to obtain information. A WSN is designed to perform a set of high-level information processing tasks such as detection, tracking, or classification. WSNs can be applied in multiple fields such as military, agriculture, traffic, industry, and environmental protection, and it is one of the most important research topics in computer fields [1]. There should be one or a few sink nodes and a number of sensor nodes in WSNs. WSNs should be having a contact with a base station. All sink nodes communicate with base station. Sink node's energy is supplied by cable, and it should be unlimited; sensor

node's energy is supplied by battery and it is limited. If some sensor nodes' energy is exhausted, information from the area monitored by these nodes will not be obtained. And dead nodes will not relay data from other nodes; thus, other sensor nodes will be increasingly burdened with transmission. Given these issues, energy consumption in WSNs is an important research spot. So raising the sensor node's energy efficiency is an important factor to improve the performance of WSNs. Many researches focus on modifying the sensor node's energy efficiency, and designing an efficient routing algorithm is one of the most important approaches [2].

There are three kinds of routing algorithms about energy efficiency in research: data centric routing algorithm, hierarchical routing algorithm, and location-based routing algorithm. A data centric routing algorithm improves energy efficiency by employing short metadata descriptions in advertisements of the availability of data [3]. Data centric routing algorithms find and update the routing information at the same time of transmitting the simple architecture, (ii) easy to

be deployed, (iii) strong self-organization ability, but these algorithm has a major problem such as (i) flooding mechanism to transmit information (ii) overhead control packets, this kind of routing algorithms are mostly used in small scale of WSNs [4, 5]. The location-based routing algorithm uses location information to guide routing discovery and maintenance as well as data forwarding, enabling directional transmission of the information and avoiding information flooding in the entire network. This algorithm has the advantage as the route would be found very quickly, and the routing information would be accurate, but its efficiency is highly influenced by the geographical environment of WSNs [6].

Chung Horng et al [7] discussed about the HAC Hierarchical Agglomerative Clustering for the wireless sensor network. HAC is a conceptually and mathematically simple clustering approach. HAC has two important categories, divisive and agglomerative. Data types could be either quantitative or qualitative. Resemblance coefficient also has two types, dissimilarity coefficient and similarity coefficient. An input data set for HAC is a component attribute data matrix. Components are the entities that we want to group based on their similarities. Attributes are the properties of the components. Hierarchical routing is the procedure of arranging nodes in a hierarchical manner. It is one of the most popular researches in WSNs, and many typical algorithms are proposed. In this algorithm sensors are divided into several groups based on some characteristics called as cluster. In every cluster, there would be a selected node acts as Cluster Head (CH) which is responsible for collecting data from other nodes in a cluster. All other nodes are called slaves nodes are member nodes. Every cluster head has a connection with the base node or sink node. It will manage every data collected from every cluster head data in WSN. This kind of algorithm has advantages as it is robust and strong and the energy of every node is well balanced. But this algorithm has major limitation such as selection of Cluster head.

Hierarchical agglomerative clustering (HAC) [8] is a conceptually and mathematically simple clustering approach which uses four clustering methods, CLINK, SLINK, UPGMA, and WPGMA. All of these methods comprise three common key steps: obtain the data set, build the similarity matrix, and execute the clustering algorithm. Based on the concept of HAC, DHAC

method was proposed for distributed environments by improving the HAC algorithms. The main idea behind DHAC is that a node only needs one-hop neighbor knowledge to build clusters. To apply the DHAC algorithm in WSNs, a bottom-up clustering approach is followed using simple six steps. Firstly, the qualitative connectivity data is obtained as input data set for DHAC. Secondly, the similarity matrix is built. Thirdly, the similar nodes are grouped together by executing the distributed clustering algorithm. The last three steps are cutting the cluster tree with the threshold, merging the smaller cluster, and electing the CHs. the performance of DHAC is much better than LEACH. The clustering dissipated energy of DHAC is about 4 times less than that of LEACH which predicates that DHAC can achieve much high reliability and efficiency in energy consumption at 300 rounds timing.

## II. HAC(HIERARCHICAL AGGLOMERATIVE CLUSTERING)

### 2.1 Basic Concepts of HAC Algorithm

In this information explosion age, people are overwhelmed by a huge amount of data. To analyze these data and make them more understandable, classification of data is necessary. Organizing a huge amount of data into groups can lead to discovering the patterns of these data which provide us a more comprehensive understanding on them. Clustering is one of the data classifying technique. HAC is a family of algorithms that is a subdivision of hierarchical clustering techniques based on agglomerative method. In HAC, data are not classified into certain number of groups at a single step. On the contrary, it consists of several steps, which can run from in clusters each contains one data object to one final cluster that contains all data objects [9]. In general, the original raw data will be processed first and then put into a resemblance matrix for comparison. The resemblance matrix will show the dissimilarity or similarity (depending on the way of processing data) relationship between data objects. Then the data objects are classified into clusters according to the resemblance matrix, and these clusters will become larger through the multistep operation. A dendrogram can be created along with this multistep operation which can provide a visualization of the relationship among these data objects, the visualized relationship will be more distinct if hierarchical relationships exist [10], Since HAC is an

uncomplicated and efficient data analysis algorithm, it has been applied to many fields.

## 2.2 Operation of HAC Algorithm

There are three essential procedures while operating HAC algorithm, which are: obtain input data set, calculate resemblance coefficients and execute HAC algorithm [11]. During the process, various types of data, equations and methods can be applied. The input data can be either quantitative or qualitative, and different types of data need to be processed by different kinds of equation to calculate resemblance coefficients. Further, there are several methods for clustering.

### 2.2.1. Step 1: obtain the input data set

An input data set for HAC is a component–attribute data matrix. Components are the nodes that we want to group based on their similarities. Nodes exchange HELLO messages and obtain neighbor nodes’ attributes. Attributes are the properties of the components such as the location of nodes, the RSS, the connectivity of nodes, or other features. The type of input data set can be classified into quantitative data and qualitative data. Figure 1, shows a randomly generated 8-node network in the 10x10 m<sup>2</sup> field. As illustrated in Table 1a, the location information is used as the quantitative input data. Table 1b uses the one-hop network connectivity data as the qualitative input data, where the “1” value represents a one-hop connection and the “0” value represents no direct connection.

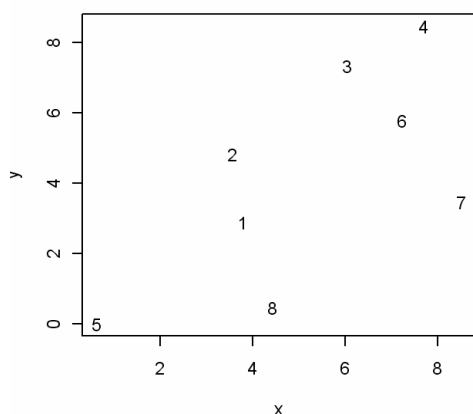


Figure 1: A simple 8-node network.

Table 1: Node input data matrix for the 8-node network.

Component (node)	Attribute							
	x-Axis	y-Axis						
<i>(a) Quantitative data: node location data matrix</i>								
(1)	3.78	2.9						
(2)	3.56	4.83						
(3)	6.06	7.34						
(4)	7.71	8.46						
(5)	0.63	0.01						
(6)	7.23	5.78						
(7)	8.52	3.46						
(8)	4.43	0.48						
	(1) (2) (3) (4) (5) (6) (7) (8)							
<i>(b) Qualitative data: one-hop network connectivity data matrix</i>								
(1)	1	1	0	0	1	0	0	1
(2)	1	1	1	0	0	1	0	0
(3)	0	1	1	1	0	1	0	0
(4)	0	0	1	1	0	1	0	0
(5)	1	0	0	0	1	0	0	1
(6)	0	1	1	1	0	1	1	0
(7)	0	0	0	0	0	1	1	0
(8)	1	0	0	0	1	0	0	1

### 2.2.2. Step 2: compute the resemblance coefficients

A resemblance coefficient for a given pair of nodes indicates the degree of similarity or dissimilarity between these two nodes. It could be quantitative (e.g., location, RSS) or qualitative (e.g., connectivity). We can calculate Euclidean distance based on the location information by using the Pythagorean Theorem. In Eq. (1), x and y represent the location of node, a and b, on x-axis and y-axis.

$$\text{Euclidean distance: } D_{ab} = [(a_x - b_x)^2 + (a_y - b_y)^2]^{1/2}$$

To deal with the qualitative data, there are various ways to calculate the resemblance coefficients [12]. There are three typical methods:

- 1) Jaccard Coefficient:  $c_{xy} = a / (a + b + c)$
- 2) Simple Matching Coefficient:  $c_{xy} = (a + d) / (a + b + c + d)$
- 3) Sorenson Coefficient:  $c_{xy} = 2a / (2a + b + c)$

### 2.2.3. Step 3: execute the HAC algorithm

After building up the resemblance matrix, the HAC algorithm repeatedly identifies the minimal coefficient in the resemblance matrix and executes the clustering algorithm to assign the nodes into a “tree”. Each step includes merging two clusters together and updating the resemblance matrix. Updating the resemblance matrix is an important step, and various algorithm methods can be adopted. There are four main types of the HAC algorithm methods [3].

**Table 2:** Resemblance matrix with quantitative data using Euclidean distance

	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	1.94	4.99	6.81	4.27	4.49	4.77	2.51
(2)	-	3.54	5.51	5.64	3.79	5.15	4.44
(3)	-	-	1.99	9.12	1.95	4.59	7.05
(4)	-	-	-	11	2.72	5.07	8.63
(5)	-	-	-	-	8.77	8.61	3.83
(6)	-	-	-	-	-	2.65	5.99
(7)	-	-	-	-	-	-	5.06

- Single LINKage (SLINK), also called the nearest neighbor method. It defines the similarity measure between two clusters as the minimum resemblance coefficient among all pair entities in the two clusters.

$$C_{\text{slink}} = \text{Min} (C_{(1,1)}, C_{(1,2)}, \dots, C_{(i,j)}, \dots, C_{(m,n)})$$

- Complete LINKage (CLINK), also called the furthest neighbor method. It defines the similarity measure between two clusters as the maximum resemblance coefficient among all pair entities in the two clusters.

$$C_{\text{clink}} = \text{Max} (C_{(1,1)}, C_{(1,2)}, \dots, C_{(i,j)}, \dots, C_{(m,n)})$$

- Un-weighted Pair-Group Method using arithmetic Averages (UPGMA). This defines the similarity measure between two clusters as the arithmetic average of resemblance coefficients among all pair entities in the two clusters. UPGMA is the most commonly adopted clustering method.

$$C_{\text{UPGMA}} = 1/mn \sum_{i=1, j=1}^{m,n} C_{(i,j)}$$

- Weighted Pair-Group Method using arithmetic Averages (WPGMA). This is the simple arithmetic average of resemblance coefficients between two clusters without considering the cluster size.

$$C_{\text{WPGMA}} = 1/mn \sum_{i=1, j=1}^{m,n} C_{(i,j)} W_i$$

The results of the HAC algorithm are usually depicted by a binary tree or dendrogram, as shown in Fig. 2. The root node of the dendrogram represents the whole data set and each leaf is regarded as a node. The intermediate nodes, thus, describe the extent that the nodes are proximal to each other and the height of the dendrogram. Figure 2 demonstrates three different clustering results for the simple 8-node network depicted in Figure 1 by using SLINK, CLINK, UPGMA methods with quantitative data.

## 2.2.4. Step 4: cut the hierarchical cluster tree

To avoid clusters become oversized and to stop merging of clusters, we make a cut by using a pre-configured threshold value, such as transmission radius, number of clusters, or cluster density. Figure 3b shows a cutting of transmission radius basing on the clustering result of UPGMA with quantitative data. Figure 4 illustrates that the sample network has three corresponding clusters, {3, 6, 4, 7}, {1, 2, 8}, and {5} based on Figure 3b.

**Table 3:** Resemblance matrix with qualitative data using SORENSON dissimilarity

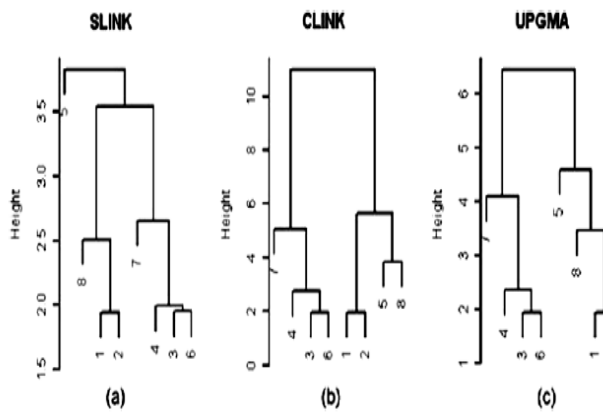
	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	0.5	0.75	1	0.143	0.778	1	0.143
(2)	-	0.25	0.429	0.714	0.333	0.667	0.714
(3)	-	-	0.143	1	0.111	0.667	1
(4)	-	-	-	1	0.25	0.6	1
(5)	-	-	-	-	1	1	0
(6)	-	-	-	-	-	0.429	1
(7)	-	-	-	-	-	-	1

## Step 5: control the minimum cluster size

If the size of a cluster is smaller than the predefined threshold, minimum cluster size, the cluster merges with its closest neighboring cluster. Figure 3 shows that the small Cluster {5} is merged into Clusters {1, 2, 8}. Thus, Figure 5 presents the final formatted clusters {3, 6, 4, 7} and {1, 2, 8, 5}.

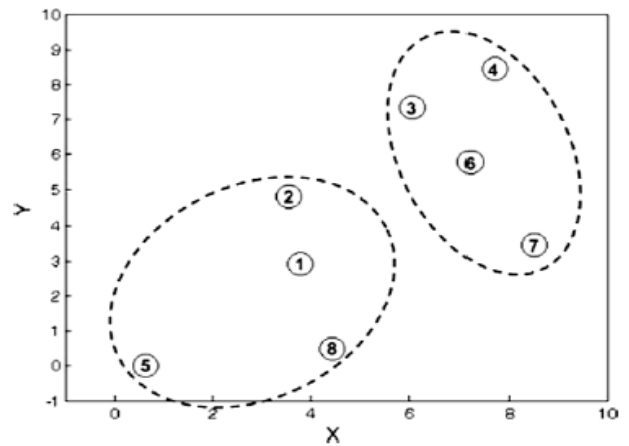
## 2.2.5. Step 6: choose CHs

Once clustering is finished, CHs can be initially determined by different strategies.



**Figure 2 :** Dendrogram using different the HAC algorithms with quantitative data

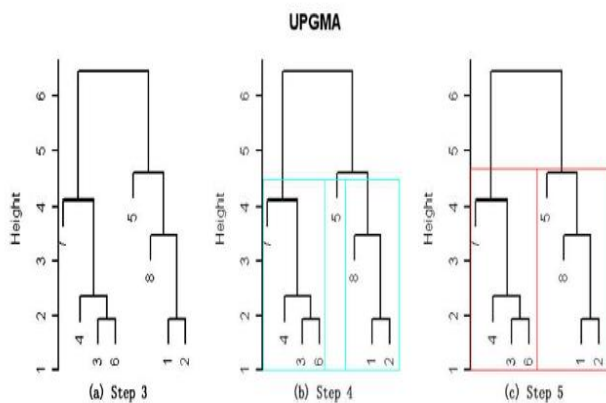
In this paper, CHs are the nodes that satisfy two conditions: (i) the node is in the bottom level, which merged into the cluster in the first step and (ii) the node has the lower ID. As demonstrated in Figure 5, nodes 1 and 6 are the CHs of the Cluster {1, 2, 8, 5} and {3, 6, 4, 7}, respectively.



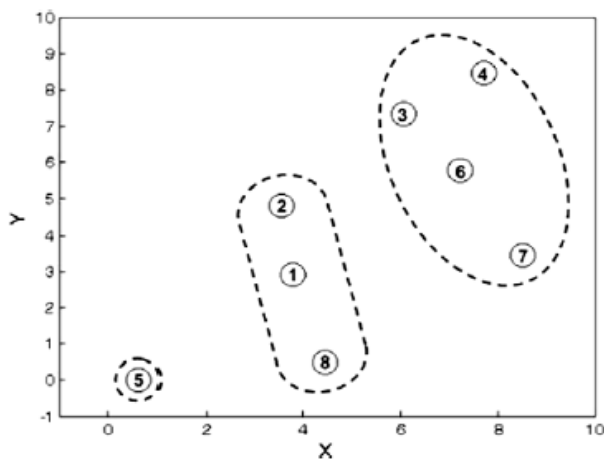
**Figure 5:** Generated clusters at Step 5, using UPGMA with quantitative data.

### III. DISTRIBUTED HIERARCHICAL AGGLOMERATIVE CLUSTERING (DHAC)

The idea of DHAC is that for WSNs, the HAC algorithm does not actually need the global knowledge. To be specific, a node can make use of only one-hop neighbor knowledge for the computation. Nodes that are far apart will not be grouped into the same cluster anyway. Therefore, it is not necessary to include all the information for clustering. As a result, DHAC can remove the assumption that each node has the global information of every other node. We present the DHAC with the concept of distributed clustering for WSNs based on the quantitative location data and qualitative connectivity knowledge using the different methods. DHAC adopts the “general assumptions” of WSNs as follows:



**Figure 3:** Clustering steps and dendrogram using UPGMA with quantitative data



**Figure 4:** Generated clusters at Step 4, using UPGMA with quantitative data

- The nodes in the network are quasi-stationary.
- The nodes are left unattended after deployment.
- Each node only has local information or the identification of its one-hop neighbor nodes.
- All nodes have similar capabilities, processing, communication and initial energy.
- Propagation channel is symmetric.
- The transmission ranges of nodes are adjustable. All the nodes have the capability to communicate with the sink directly.
- The sink is static.

#### 3.1. DHAC: cluster formation

Without the global knowledge, DHAC can make use of the neighboring information to determine if a node actually needs to perform the clustering task. The



rationale is that every node knows its one-hop neighbors. Figure 6, illustrates the pseudo code of the DHAC implementation for WSNs.

### 3.1.1. Step 1 and 2: obtain input data set and build resemblance matrix

The procedure ‘‘Set up ResM’’ (Figure 6, lines 1–7) corresponds to ‘‘obtain input data set’’ and ‘‘build resemblance matrix’’ steps are described. To collect input data and set up the local resemblance matrix, in the beginning, each node elects itself as a cluster head and exchanges the information via HELLO messages with its neighbors. In Figure 6, lines 1–7 initialize the clustering process by setting the current  $ID_{CH}$  as  $ID_{Node}$  and exchanging HELLO messages with one-hop neighbors.

### 3.1.2. Step 3: execute the DHAC algorithm

After the clustering process ends, each cluster establishes its own local resemblance matrix, from which its minimum coefficient ( $M_{C_{coeff}}$ ) can be easily found. Each cluster then determines its minimum cluster head ( $M_{CH}$ ). If the  $ID_{Node}$  of the  $M_{CH}$  is larger than the CH, the CH will send an INVITE message to the  $M_{CH}$ . In Figure 6, lines 10–16 specify two requirements: the sender node must be the current CH, and the sender  $ID_{Node}$  must be smaller than  $M_{CH}$ . The following illustrates DHAC in detail, step by step:

<b>Definitions:</b>	
$ID_{Node}$ :	Node ID
$ID_{CH}$ :	Cluster Head ID
ResM :	Resemblance Matrix
$M_{C_{coeff}}$ :	The Minimum_coefficient in the ResM
$M_{CH}$ :	The $ID_{CH}$ of the cluster corresponds to the $M_{C_{coeff}}$ in the ResM
$C_{Size}$ :	The number of cluster_member in a cluster
1: <b>procedure</b> SET UP RESM -	
exchange information with neighbors	
2: $ID_{CH} \leftarrow ID_{Node}$	
- self form a singleton cluster	
3: Send HELLO message to one-hop neighbors	
4: .....	
- keep listening from neighbors	
5: Receive HELLO messages	
6: Establish ResM	
7: <b>end procedure</b>	
8: <b>procedure</b> EXECUTE DHAC	
9: <b>while</b> $M_{C_{coeff}} < Threshold$ <b>do</b>	

10: <b>if</b> $ID_{CH} == ID_{node}$ <b>then</b>	-Sending message
11: $M_{C_{coeff}} \leftarrow$ minimum value in ResM	
12: $M_{CH} \leftarrow ID_{CH}$ With $M_{C_{coeff}}$	
13: <b>if</b> $ID_{CH} < M_{CH}$ <b>then</b>	
14:     Send INVITE message to $M_{CH}$	
15: <b>end if</b>	
16: <b>end if</b>	
17: <b>if</b> Received INVITE message <b>then</b>	
18: <b>if</b> $M_{CH} =$ Sender $ID_{CH}$ <b>then</b>	
19:         send CONFIRM message	
20: $ID_{CH} \leftarrow$ Sender $ID_{CH}$	
21: <b>else</b>	
22:         Send REJECT message	
23: <b>end if</b>	
24: <b>else if</b> Received CONFIRM message <b>then</b>	
Go To MERGE.CLUSTERS	
25: <b>else if</b> Received INFORM message <b>then</b>	
Update ResM	
26: <b>else if</b> Received REJECT message <b>then</b>	
Stop send INVITE message	
until ResM updated	
27: <b>end if</b>	
28: <b>end while</b>	
29: <b>end procedure</b>	
30: <b>procedure</b> MERGE_CLUSTERS -	
cluster merging procedure	
31: Merge two clusters	
32: Combine neighboring information of two clusters	
33: Update ResM by using chosen method	
34: Broadcast INFORM message	
35: <b>end procedure</b>	

Figure 6: Pseudo code of DHAC in WSNs

- In Figure 6, lines 17–23, when a CH receives an INVITE message, it checks the source of the message. If the source is its MCH, the CH sends a CONFIRM message back to the source, selects the source to be the new CH, and turns into the sleep mode; otherwise, the CH sends back a REJECT message. For example, node 2 is the initial CH in Cluster {2}. When receiving an INVITE message from Cluster {1}, Cluster {2} sends a CONFIRM message back to Cluster {1}.
- Once a CH receives a REJECT message from its MCH, the CH stops sending the INVITE message until its resemblance matrix has been updated. For example, Cluster {5} will stop sending the invite message to Cluster {8}.
- After receiving a CONFIRM message, the CH updates its neighbor list and resemblance matrix, then it merges with the cluster headed by MCH to form a new cluster (Figure 6, line 33–38). The resemblance matrices of two clusters are updated through the chosen HAC algorithm. Clusters {1}

and {2}, for instance, will merge their resemblance matrices, and their neighbor lists.

- The CH of the new cluster broadcasts an INFORM message to notify its neighbors to update their resemblance matrices (Figure 6, lines 25 and 37). Clusters update their own resemblance matrix after receiving this INFORM message, which contains the new cluster information and the merged neighbor list.

The process will repeat until the while condition (line 9) fails. During the clustering process, all CHs of clusters keep listening. When a CH receives a message, it reacts based on the message type.

### 3.1.3. Step 4: cut the hierarchical cluster tree

Using a predefined threshold, the while loops in Figure 6, lines 9–28, controls the upper bound size of clusters. The control conditions correspond to the step of cutting the hierarchical cluster tree.

### 3.1.4. Step 5: control the minimum cluster size

After clusters are generated by running DHAC, the minimum cluster size can also be used to limit the lower bound of cluster size by performing the procedure “MERGE CLUSTERS” (Figure 6, lines 29–31).

### 3.1.5. Step 6: choose CHs

To choose the corresponding CHs, DHAC simply choose the lower ID node between the two nodes that join the cluster at the first step. The CH chosen does not require extra processing.

## IV. CONCLUSION

Clustering has been a topic of interest in many different disciplines for a long time. To adapt to the constraints of WSNs, clustering has generated a lot of discussion. This paper advocated the application of well-known the HAC algorithm methods to WSNs and another distributed approach, DHAC algorithm, to classify sensor nodes into appropriate groups instead of simply gathering nodes by their distance to some randomly selected CHs. In this paper demonstrated the well-understood HAC algorithm methods, SLINK, CLINK, UPGMA and WPGMA. Based on the study of the HAC algorithm methods, we illustrated how to use the DHAC approach to mitigate the problems encountered

with current protocols, and we supported the analysis by simulation.

## V. REFERENCES

- [1]. Stallings W. Wireless communications and networks. 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall; 2005.
- [2]. Liu B. Unsupervised Learning. Web Data mining exploring hyperlinks, contents and usage data. 2nd ed. Springer; 2010. p. 134–55.
- [3]. Anderberg MR. Cluster Analysis for Applications. New York: Academic Press; 1973.
- [4]. Heinzelman WR, Kulic J, Balakrishnan H. Adaptive protocols information in wireless sensor networks. Proceedings of the 5th ACM/IEEE MOBICOM; 1999.
- [5]. Lung CH, Zhou CJ. Using hierarchical agglomerative clustering in wireless sensor networks: an energy-efficient and flexible approach. Ad Hoc Networks. 2010; 8(3):328–44.
- [6]. Ho JH, Shih HC, Liao BY, Chu SC. A ladder diffusion algorithm using ant colony optimization for wireless sensor networks. Information Sciences. 2012; 192(1):204–12.
- [7]. Lung C-H, Zhou C, Yang Y. Applying hierarchical agglomerative clustering to wireless sensor networks. Ad Hoc Networks. 2004; 8(3):328–44.
- [8]. Chung-Horng Lung, Chenjuan Zhou, “Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach” IEEE "GLOBECOM" 2008 proceedings.
- [9]. B. S. Everitt, S. Landau, M. Leese, and D. D. Stahl, *Cluster analysis*, 5 ed.: WILEY, 2010.
- [10]. X. Rui and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, pp. 645-678, 2005.
- [11]. C. Zhou, "Application and Evaluation of Hierarchical Agglomerative Clustering in Wireless Sensor Networks," MASc Thesis, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 2008.
- [12]. S. S. Choi, S. H. Cha, and C. Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal on Systemics, Cybernetics and Informatics*, vol. 8, pp. 43-48, 2010.
- [13]. M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.
- [14]. Pal N.R, Pal K, Keller J.M. and Bezdek J.C, “A Possibilistic Clustering Algorithm”, *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 4, Pp. 517–530, 2005.
- [15]. R. Krishnapuram and J.M. Keller, “A possibilistic approach to clustering”, *IEEE Trans. Fuzzy Systems*, Vol. 1, Pp. 98-110, 1993.

- [16]. J. C. Dunn (1973): "A Agglomerative Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: 32-57

## VI. AUTHORS PROFILE



**S.Aravindhana** received his M.Phil degree from Thiruvalluvar university, Vellore in the year 2012. He has received his MCA degree from Anna University, Chennai in the year 2011. He is pursuing his Ph.D degree at Sri Vijay Vidyalaya College of Arts & Science (Affiliated Periyar University), Dharmapuri, Tamilnadu, India. His areas of interest include Data Mining, Cloud Computing and Computer Networks.



**Dr.D.Maruthanayagam** received his Ph.D Degree from Manonmaniam Sundaranar University, Tirunelveli in the year 2014. He received his M.Phil Degree from Bharathidasan University, Trichy in the year 2005. He received his M.C.A Degree from Madras University, Chennai in the year 2000. He is working as HOD Cum Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. He has 15 years of experience in academic field. He has published 4 books, 22 papers in International Journals and 28 papers in National & International Conferences so far. His areas of interest include Computer Networks, Grid Computing, Cloud Computing and Mobile Computing.