The K-Means Clustering used in Wireless Sensor Network

Dr Gayatri Devi
Professor
Department of Computer Science and Engineering
ABIT,Sec-1,CDA, Cuttack, Odisha, India
gayatridevi13@yahoo.com

Srutipragyan Swain
Lecturer
Department of Computer Science and Engineering
IMIT,Jobra, Cuttack, Odisha, India
sruti56@gmail.com

Mr Rajeeb Sankar Bal
Senior Lecturer
Department of Computer Science and Engineering
ABIT,Sec-1,CDA, Cuttack, Odisha, India
rajiv.s.bal@gmail.com

Abstract—The past few years have witnessed increased interest in the potential use of wireless sensor networks in applications such as environment management and various surveillance. The Sensor nodes in these applications are expected to be remotely deployed in large numbers and to operate autonomously in unattended environments. As per scalability, the nodes are often grouped into disjoint and mostly non-overlapping clusters. We propose K-mean clustering used wireless sensor network. The method can divide a sensor network into a few clusters.

Keywords- *Wireless Sensor Network (WSN); Wireless Sensor (WS); Sensor Node (SN).*

I. INTRODUCTION

The WSNs or sensor network have been widely considered as one of the most important technologies for the twenty first century shown in Fig 1. A sensor network is an infrastructure comprised of sensing (measuring), computing, and communication elements that gives an administrator the ability to instrument,observe, and react to events and phenomena in a specified environment. The administrator typically is a civil, governmental, commercial, or industrial entity.The environment can be the physical world, a biological system, or an information technology (IT) framework. The sensing is often also interested in control and activation.There are four basic components in a sensor network: (1) an assembly of distributed or localized sensors; (2) an interconnecting network (usually, but not always, wireless-based); (3) a central point of information clustering; and (4) a set of computing resources at the central point (or beyond) to handle data correlation, event trending, status querying, and data mining.The computation and communication infrastructure associated with sensor networks is often specific to this environment and rooted in the device and application-based nature of these networks. For example, unlike most other settings,in-network processing is desirable in sensor networks; furthermore, node power (and/or battery life) is a key design consideration. The information collected is typically parametric in nature, but with the emergence of low-bit-rate video [e.g., Moving Pictures Expert Group 4 (MPEG-4)] and imaging algorithms, some systems also support these types of media.

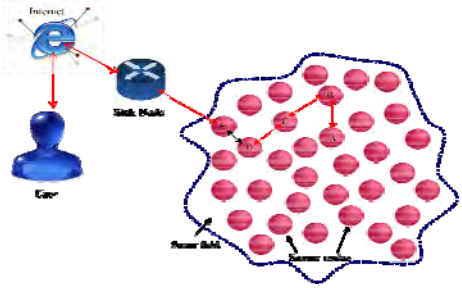

Figure 1. The computation and communication architecture of WS or WSN.

A. WSN Vs. Ad-Hoc Network

In Figure.2, a mobile ad hoc network (MANET), sometimes called a mobile mesh network, is a self configuring network of mobile devices connected by wireless links [1-3]. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. The difference between wireless sensor networks and ad-hoc networks are outlined below:

- The number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad hoc network.
- Sensor nodes are densely deployed.
- Sensor nodes are prone to failures.
- The topology of a sensor network changes very frequently.
- Sensor nodes mainly use broadcast communication paradigm whereas most ad hoc networks are based on point-to-point communication
- Sensor nodes are limited in power, computational capacities, and memory.
- Sensor nodes may not have global identification (ID) because of the large amount of overheads and large number of sensors.
- Sensor networks are deployed with a specific sensing application in mind whereas adhoc networks are mostly constructed for communication purpose.

In [2], we face in designing sensor network systems and applications include:-

- Limited hardware: Each node has limited processing, storage, and communication capabilities, and limited energy supply and bandwidth.
- Limited support for networking: The network is peer-to-peer, with a mesh topology and dynamic, mobile, and unreliable connectivity. There are no universal routing protocols or central registry services.
- Limited support for software development: The tasks are typically real-time and massively distributed, involve dynamic collaboration among nodes.

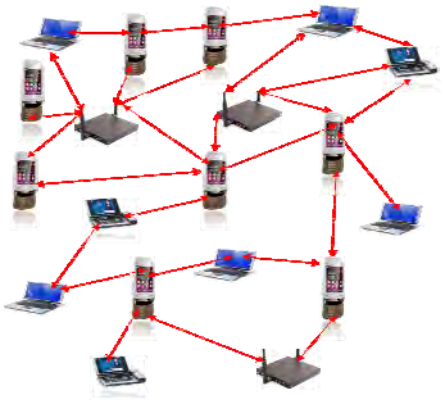

Figure 2. The overall view of Mobile Ad-hoc Network.

B. The WSN Applications

WSNs may consist of many different types of sensors including seismic, magnetic, thermal, visual, infrared, acoustic, and radar, which are able to monitor a wide variety of ambient conditions that include temperature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, the presence or absence of certain kinds of objects, and mechanical stress levels on attached objects. As a result, a wide range of applications are possible. This spectrum of applications includes homeland security, monitoring of space assets for potential and human-made threats in space, groundbased monitoring of both land and water, intelligence gathering for defense, environmental monitoring, urban warfare, weather and climate analysis and prediction, battlefield monitoring and surveillance, exploration of the Solar System and beyond, monitoring of seismic acceleration, strain, temperature, wind speed and GPS data. These ever-increasing applications of WSNs can be mainly categorized into five categories shown in figure 3.

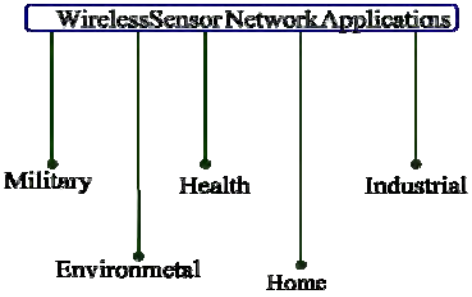

Figure 3. The main categories of WSN applications.

II. THE CLUSTERING IN WSN

A. Clustering of Sensors:

In the clustering of Sensors, it is clear that enough number of SNs need to be deployed if every corner of the area of interest, need to sensed for continuous monitoring and necessary action. In addition, successful transfer of sensed data to adjacent SNs necessitates minimum communication distance covered by the wireless radio to be at least twice that of sensing range. So, sensing distance and communication coverage are co-related. It is widely accepted that the energy consumed in one bit of data can be used to perform a large number of arithmetic operations in the sensor processor. Moreover, the way sensors are deployed, physical environment would produce similar in close by SNs and transmitting such data could be termed as more or less redundant. Therefore, all these facts encourage using some kind of grouping (clustering) so that data from SNs belonging to a single cluster can be combined together in an intelligent way (aggregation) using local transmissions to transfer only the compact data. This can not only reduce the global data to be transferred and localized most traffic to within each individual cluster, but also reduces the traffic and hence contention in a

WSN. A lot of research gone into testing coverage of areas by k-sensors (k>l) clustering adjacent SNs and defining the size of the cluster so that the cluster heads (CHs) can easily get data from their own cluster members and CHs can also communicate among themselves and exchange data. If each cluster is covered by more than one subset of SNs all the time, then some of the SNs can be put into sleep mode so as to conserve energy while keeping full coverage of each cluster and the area. The use of a second smaller radio has been suggested for waking up the sleeping sensor, thereby conserving the power of main wireless transmitter. In WSN, there are two types of clustering techniques. The clustering technique applied in homogeneous sensor networks is called homogeneous clustering schemes, and the clustering technique applied in the heterogeneous sensor networks is referred to as heterogeneous clustering schemes[4].

III. THE K-MEANS CLUSTERING USED IN WSN

A. K-MEANS clustering Algorithm

A k-means algorithm is outlined below There are several ways to select the initial k points that represent the clusters.The heart of the algorithm is the for-loop, in which we consider each point other than the k selected points and assign it to the closest cluster, where “closest” means closest to the centroid of the cluster. Note that the centroid of a cluster can migrate as points are assigned to it. However, since only points near the cluster are likely to be assigned, the centroid tends not to move too much.

How the K-Mean Clustering algorithm works?

If the number of data is less than the number of cluster then we assign each data as the centroid of the cluster.

Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, we calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data.Since we are not sure about the location of the centroid, we need to adjust the centroid location based on the current updated data. Then we assign all the data to this new centroid. This process is repeated until no data is moving to another cluster anymore.

Initially choose k points that are likely to be in different clusters;

Make these points the centroids of their clusters;

FOR each remaining point p DO

find the centroid to which p is closest;

Add p to the cluster of that centroid;

Adjust the centroid of that cluster to account for p;

END;

Initializing Clusters for K-Means

We want to pick points that have a good chance of lying in different clusters.

There are two approaches.

Pick points that are as far away from one another as possible.

Cluster a sample of the data, perhaps hierarchically, so there are k clusters. Pick a point from each cluster, perhaps that point closest to the centroid of the cluster.

The second approach requires little elaboration. For the first approach, there are variations. One good choice is:

Pick the first point at random;

WHILE there are fewer than k points DO

Add the point whose minimum distance from the selected points is as large as possible;

END;

Example: Let us consider the twelve points of Fig.4.In the worst case, our initial choice of a point is near the center, say (6, 8). The farthest point from (6, 8) is (12, 3), so that point is chosen next.

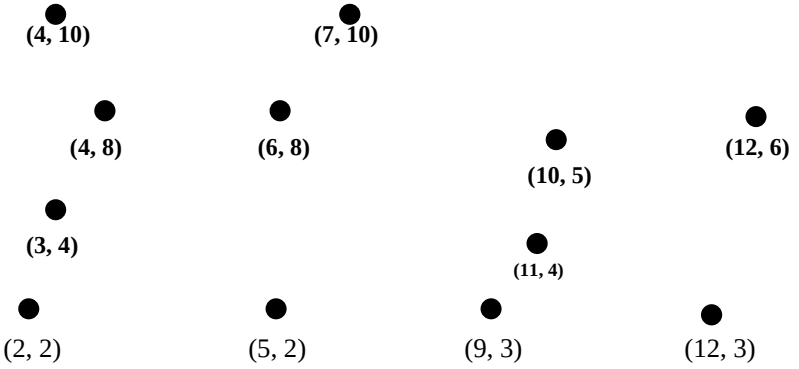

Figure 4: Twelve points to be clustered

Among the remaining ten points, the one whose minimum distance to either (6,8) or (12,3) is a maximum is (2,2). That point has distance √52 = 7.21 from (6,8) and distance √101 = 10.05 to (12,3); thus its "score" is 7.21. You can check easily that any other point is less than distance 7.21 from at least one of (6,8) and (12,3). Our selection of three starting points is thus (6,8), (12,3), and (2,2). Notice that these three belong to different clusters. If we start with a different point, say (10,5), we would get a different set of three initial points. In this case, the starting points would be (10,5), (2,2), and (4,10). Again, these points belong to the three different clusters.

B. Picking The Right Value of K

We may not know the correct value of k to use in a k-means clustering. However, if we can measure the quality of the clustering for various values of k, we can usually guess what the right value of k is. In the above example, we observed that if we take a measure of appropriateness for clusters, such as average radius or diameter, that value will grow slowly, as long as the number of clusters we assume remains at or above the true number of clusters. However, as soon as we try to form fewer clusters than there really are, the measure will rise precipitously. The idea is expressed by the below diagram:

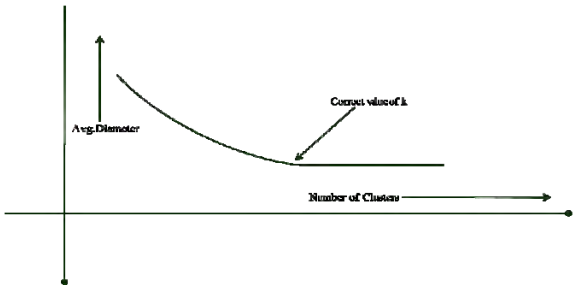

Figure 5: The Correct value of k in a graph of Average Diameter Vs. Number of Clusters

Figure 5. Average diameter or another measure of diffuseness rises quickly as soon as the number of clusters falls below the true number present in the data. If we have no idea what the correct value of k is, we can

find a good value in a number of clustering operations that grows only logarithmically with the true number. Begin by running the k-means algorithm for k = 1, 2, 4, 8, ... Eventually, you will find two values v and 2v between which there is very little decrease in the average diameter, or whatever measure of cluster cohesion you are using. We may conclude that the value of k that is justified by the data lies between v/2 and v. If you use a binary search (discussed below) in that range, you can find the best value for k in another log2 v clustering operations, for a total of 2 log2 v clusterings. Since the true value of k is at least v/2, we have used a number of clusterings that is logarithmic in k. Since the notion of "not much change" is imprecise, we cannot say exactly how much change is too much. However, the binary search can be conducted as follows, assuming the notion of "not much change" is made precise by some formula. We know that there is too much change between v/2 and v, or else we would not have gone on to run a clustering for 2v clusters. Suppose at some point we have narrowed the range of k to between x and y. Let z = (x + y)/2. Run a clustering with z as the target number of clusters. If there is not too much change between z and y, then the true value of k lies between x and z. So recursively narrow that range to find the correct value of k. On the other hand, if there is too much change between z and y, then use binary search in the range between z and y instead.

C. Applications of K-mean clustering

There are a lot of applications of the K-mean clustering, range from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligent, image processing, machine vision, etc. In principle, you have several objects and each object have several attributes and you want to classify the objects based on the attributes, then you can apply this algorithm.

IV. SIMULATION

- It generates the sensor nodes randomly & each sensor node contains x-axis & y-axis coordinate system using the java program in figure 6 .

```
F:\k-means-java>javac MainClass.java

F:\k-means-java>java MainClass
Enter no. of nodes=
9
Nodes are
Node1 : 88     13
Node2 : 19     63
Node3 : 9      80
Node4 : 84     26
Node5 : 44     55
Node6 : 14     37
Node7 : 27     72
Node8 : 43     72
Node9 : 56     79
```

Figure 6: Randomly generated sensor nodes with x-coordinate and y-coordinate.

Now we are generating the sensor nodes in WSN shown in figure 7.

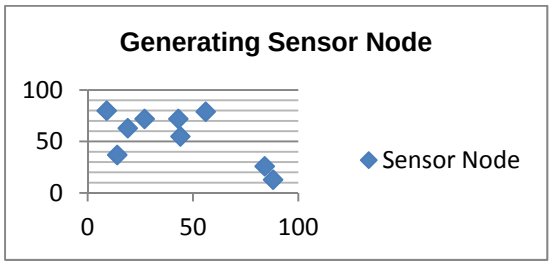

Figure 7: The graph of randomly generated sensor nodes.

The Sensor nodes are forming Clustering shown in figure with result in figure.7 and 8:

Figure 7: The forming clustering for sensor nodes in WSN.

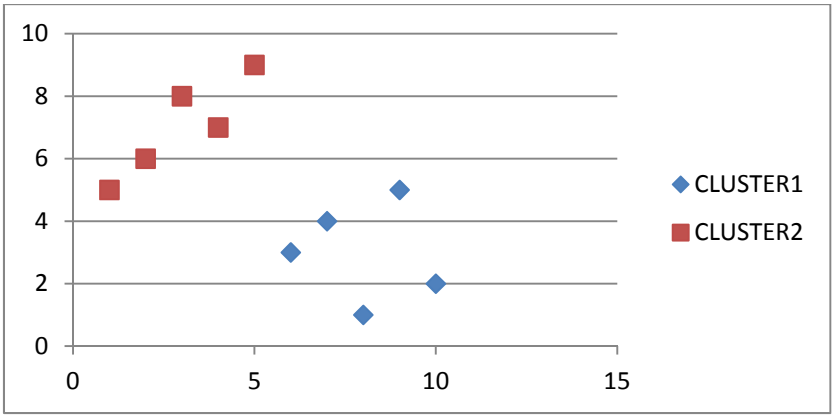

Figure 8: The separate the clustering for sensor nodes in WSN.

V. CONCLUSION

According to simulation above, the But K-means clustering algorithm has biggest advantage of clustering large data sets and its performance increases as number of clusters increases and the performance of K- mean algorithm is better than Hierarchical Clustering Algorithm. Some of the further work that is also done for further research work is: a) Different WSN is made up for different purposes. So the effect of parameters will vary from network to network. In our experiment we tried to work with those parameters which are common for almost all the sensors. But finding the right combination of parameters and their optimum value for specific WSN is a great challenge. b) For our experiment we have worked with uniform distributed data type. Further experiment can be done by choosing exponential, Negative exponential distribution.

REFERENCES

[1] I.F. Akyildiz et al., Wireless sensor networks: a survey, Computer Networks 38(2002) 393–422.
[2] C-Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8) (2003) 1247–1256.
[3] D. Estrin, et al., Next century challenges: scalable coordination in sensor networks, in: Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCom ’99), Seattle, Washington, August 1999.
[4] http://vlab.amrita.edu/?sub=78&brch=256&sim=1558&cnt=1.