 ISSN: 2321-8169
990– 996

Load Balanced With Distributed Self Organization in File Sharing and File Accessing

Ms. A. Muthulakshmi[1]

Department of Computer Science,Thiruvalluvar college of Engg & Tech
muthula277@gmail.com

Mr. S. Hariraman[2]

Assistant Professor Computer Science and Engineering
Department of Computer Science,Thiruvalluvar college of Engg & Tech
hariraman.vs@gmail.com

Abstract**—**Load balancing is a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server process..Assuming that the residual energy of nodes follows the random distribution, we propose a load-balanced clustering algorithm for WSNs on the basis of their distance and density distribution. Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage. Load balancing gives the IT team a chance to achieve a significantly higher fault tolerance. It can automatically provide the amount of capacity needed to respond to any increase or decrease of application traffic.

Keywords**—** *connectivity density, life cycle, random distribution, wireless sensor networks.*

*****

1. Introduction

The Development of tiny, low-cost and intelligent sensors with communication capabilities has prompted the emergence of wireless sensor networks (WSNs). Since, WSNs have attracted considerable attention because of their extensive applications in many areas such as object tracking, intrusion detection, environmental monitoring, traffic control, inventory management in factory and health related applications and so on. In WSNs, the energy of network node is often limited, so the efficient use of energy is a must in topology control. At the same time, since there are usually a large number of nodes in WSNs, the node can only get part of the network topology information. As a result, the clustering algorithms are needed to select the appropriate sub-cluster in local network on the basis of the partial information

The topology of WSNs often changes dynamically for a variety of reasons, including sensor node failure as battery runs out, destruction of environmental factors, location change of sensor nodes wireless communication link signal alteration due to node power control or environmental factors, the adding of new nodes into the network to enhance the monitoring accuracy, etc. Thus, WSNs should be able to get adapted to these variations and reconfigure to satisfy user's task dynamically. Meanwhile, a big concern in WSNs is the acquisition of regional data information rather than specific node information, and thus the topology control mechanism of traditional networks is frustrated. Due to energy constraints, a sensor node can but communicate directly with other sensors within a limited distance.

In order to enable communication between sensors out of each other's communication range, sensors form a multichip communication network. Utilizing clustering algorithm to form a hierarchical network topology is the accustomed realizing scenario of network management and data aggregation for WSNs, and clustering facilitates the distributed control of the network. Moreover, clustering nodes into groups not only saves energy but also reduces network contention when nodes communicate their data over shorter distances to their respective cluster-heads Motivated by the former research of clustering algorithm, the objective of this paper is to study a Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks (DSBCA), which can deal with stochastic distribution of sensor nodes. In the scenario, we could calculate the radius of cluster based on distance and distribution, and take connectivity density and residual energy of nodes into account to form clusters. The remainder of this paper is organized as follows. Section 2 reviews several cluster based algorithms proposed previously. In Section 3, we put forward the system model of our scheme. In Section 4, we investigate our new balanced clustering algorithm. Then, the performance analysis of the proposed algorithms is presented in Section 5. Finally, conclusions are given in the last section of the paper.

2. RELATEDWORK

LEACH proposed initially is a distributed and single-hop clustering algorithm. Each clustering cycle consists of cluster forming phase and data communicating phase. Simultaneously, LEACH can guarantee not only the equal probability of each node as cluster head, but also the relatively balanced energy consumption of the network nodes. Many follow-up algorithms are put forward relying on the work of LEACH. In the cluster forming phase, the cluster head nodes will inform other nodes once they are selected. The ordinary nodes can choose which cluster to join into automatically according to their distance from head nodes. In general, the election of cluster head depends on the selected threshold (n)as follows.

Where P is the percent of cluster head nodes in all nodes, nis the token of the node, and r is the number of rounds for the election. rmod(1/p) is the number of nodes elected as cluster head in a cycle, and Gis the set of nodes not elected as a cluster head, while each node is elected as the cluster head with the same probability. In each circle, first the threshold T(n)of the node elected as cluster head is set to , and then the probability of the remaining nodes increases. Thus this process can guarantee that the node with the equal probability is elected as the cluster head.

2.1 LEACH algorithm

LEACH algorithm only considers the probability for each node to become cluster head. However, the specific geographical location of nodes is left out, so the election of cluster heads is heterogeneous, as proved in a large number of experiments. Additionally, LEACH allows building single-hop cluster, and a large number of clusters might be formed consequently,

Causing great energy consumption of communications. Further studies on the LEACH algorithm

By taking overhead of communication within a cluster into account, HEED [8] algorithm overcomes the shortcoming of unevenly distributed cluster heads as enjoyed by the LEACH algorithm. And the residual energy of node as a parameter is introduced to elect cluster head with more energy to undertake

data forwarding tasks. In initialization phase, nodes send the messages to compete with the initialized probability of CHprob. When the election of cluster head is completed, other nodes join into clusters by means of the information gathered in competing phase. Here, CH prob is described as

CHprob=max(Cprob+Eresident/Emax,pmin

whereCprob and pmin are the whole network parameters affecting the convergence speed of the algorithm, the recommended values of which are given in simulation experiments; Eresident/Emaxis the ratio of the node residual energy and initial energy.

HEED algorithm is distinguished from the LEACH in cluster head selection mechanism. Meanwhile, the speed of clustering in the HEED algorithm is also improved and the cost of communication within the cluster has been taken into account. Moreover, the residual energy of node is as an important parameter to be introduced in cluster head selection mechanism, since it makes cluster heads competent for data aggregating and forwarding to form a more rational network topology. Even so, there are also some problems in HEED algorithm. For instance, the competition of cluster head may exclude some nodes from joining into any clusters

GAF is a location-based node clustering algorithm. In this algorithm, the monitored area is divided into virtual cells, and the nodes are assigned to cells as per their geographic locations. Each cell will generate a cluster head

node periodically. Cluster head nodes remain active, while other nodes could be in sleep mode to reduce energy consumption. GAF is the routing algorithm proposed in ad hoc network at first, and the method of virtual division of the cell is used for WSNs. The main idea of GAF is to get the nodes into sleep state as long as possible to reduce energy consumption in

listening state. Though, the resource of sensor nodes is limited, and the clusters based on geographical location algorithm are exigent for sensor nodes while paying no attention to the node residual energy problems. So GAF may lead to death of nodes due to premature depletion of energy, thus incurring unstable network Topology

WCA is a classical algorithm based on node degree,the number of single-hop neighbors. The election of cluster head relies upon the factors of node degree, send-receive energy and residual energy. Meanwhile the size of cluster (the communication consumes large amounts of energy when cluster is too large) is limited in order to save energy. In contrast, the WCA clustering algorithm is more comprehensive than thepreviously proposed algorithms, and

ISSN: 2321-8169
990– 996

some experiments show that the performance is more superior. Additionally, the main drawback of WCA is that it needs to obtain the weight of the node and require each node to save all the information of nodes before initializing network, so excessive amounts of computing and communications may cause excessive consumption in clustering directly. In the process of aggregating and forwarding , the overhead may also give rise to excessive energy consumption and rapid death of cluster head node, incurring instability in the network topology.

2.2 K-clustering algorithm

K-clustering algorithm can constitute maximum-hop non-overlapping clusters with partial networks topology information rather than the whole network topology. At the same time, it can also save energy to prolong network survival time. Furthermore, owing to dynamic network topology changes, it is of significance studying clustering based on local information. Nevertheless, it doesn't consider cluster size and may form unbalanced cluster. For example, some clusters contain tremendous number of nodes, which results in too large overhead of inter communication. Different improved K-clustering algorithms have been come up with successively to fix this problem. A representative algorithm is proposed by Lin and Chu [13] with using hops as one of the constraint parameters. In this algorithm, the node is elected as cluster head randomly, and the distance between cluster members and cluster head doesn't exceed k hops. The algorithm is more effective in restricting data forwarding distance, but it still doesn't solve unbalanced clustering (excessive clustering nodes). In addition, in this algorithm, only the root of subtree knows which cluster it belongs to, while other nodes don't have this info. If the cluster head or the root of subtree node fails, it would be inefficient and unfavorable to cluster again.

3. OVERVIEW

The wireless data broadcast model has good scalability for supporting an almost-unlimited number of clients .Its main limitation lies in its sequential data access: the access latency becomes longer as the number of data items increases. If we can provide (approximate) answers to spatial queries before the arrival of related data packets, we will overcome the limitation of the broadcast model. In this paper we propose a balanced clustering algorithm with distributed self-organization for WSNs of non-uniform distribution, taking into account optimal configuration of clusters. Compared with traditional clustering algorithms, the proposed algorithm can form more stable and reasonable cluster structure, and also improve the network life cycle significantly. The simulation result shows that the algorithm is feasible and has superior performance. In addition, the scenario we propose is scalable and works for different network sizes.

3.1 DISTRIBUTED SELF-ORGANIZATION FOR WIRELESS SENSOR NETWORKS

Algorithm : DSBCV

The clustering algorithm in previous concepts is uniformly distribution without considering the distance. However in normally the nodes are usually having no specific pattern in arrangement. In this case, if the clustering algorithm doesn't take the distribution of nodes into account, using uniform clustering strategy ma y lead to unbalanced topological structure, and some nodes die rapidly because of excessive energy decline. The main purpose of DSBCA is to generate clusters with more balanced nodes and avoid creating excessive clusters with many nodes. The cluster near the servers also forward the data from further clusters (all clusters need to communicate with the servers, but the long – distance consumes time) and as we all know, too many members in cluster may bring about excessive traffic consumption in management and communication.

In DSBCA algorithm the connectivity density and the location of the node trying to build a balanced clustering structure. The DSBCA is based on the connectivity density and distance from the server. The clustering radius is calculated by response time and file distribution. If two clusters have the same connectivity density, the cluster much farther from the server has larger cluster radius. If two clusters have the same distance from the server, the cluster with the higher density has smaller cluster radius.

DSBCA clustering is uniform distribution. DSBCA forms different clustering layers in which the radius are most distant from a center clustering layers are larger, and in the same layer the clustering radius is same.

DSBCA clustering in non - uniform distribution. In non- uniform distribution, the cluster radiuses are determined by the distance from the server and connectivity density nodes. With most distant from the server and lower connectivity density, the cluster radius is larger. If the closer distance from the server and lower connectivity density, the cluster radius is smaller.

DSBCA follows a distributed approach to form hierarchical structure in self- organizing mode. DSBCA select the random nodes to start clustering process first. Then the start node calculates its connected density and distance from the server. First, we find the distance from the server, the connectivity density of node and the node parameters. Finally we find the cluster head, the weight of the node is calculated by connection density.

The cluster sends messages to neighbors. The node with the highest weight will become the cluster head. It asks other nodes to join in the cluster. It contains the head message. The head message includes id of the cluster head node, id of the sending node and the number of hops from the cluster head. In DSBCA algorithm if the node does not receive any head message, it declare itself the cluster head. DSBCA set the cluster size. The number of cluster node cannot exceed the DSBCA cluster size.

3.2.Cluster Head Selecting Phase

DSBCA follows a distributed approach to establish hierarchical structure in self-organizing mode without centralcontrol. DSBCA selects the random nodes to trigger clustering process first. Then the trigger nodeUt calculates its connected density and distance from the base station to determine cluster radiuskby (5), and becomes the temporary cluster head

k=floor [β D(Ut)/Dk (Ut)]

Where D(u)is the distance from the base station of u, D k(u)is the connectivity density of node u, β is the sensor parameters determined by specific applications of WSNs, andflooris the calculation of rounding

Where RSSI is received signal strength indicator, and A is the signal strength with 1 meter distance from the base station

N k(u)is k-hop neighbors of node u

Nk(u)={v∈V|v =u∧d(u,v)≤k}

Initial energy of node u, H(u)is the times of the node u being elected as cluster head. In this way we decrease the prospects of u being elected as cluster head to balance the overall energy consumption.

In the initial stage, the node Ut triggers the clustering process and sends Hello messages to its k-hop neighbors. The neighbors ink-hop utilize (9) to calculate the respective weight, and then the node with the highest weight will become the cluster head. From then on, cluster head node broadcasts (Head _message) in its k-hop neighbors to declare itself as cluster head and asks them to join the cluster. Head _message includes the ID of cluster head node (HID), the ID of the sending node (SID) and the number of hops from the cluster head (HD). When a node receives Head _message, SID can be used to maintain a path to reach the cluster head. The algorithm discards broadcast package when HD is over k to ensure that the cluster is no more thank-hop. When a neighbor node receives Head _message, even if it is already in a cluster, it sends Join_ message to the cluster head to request joining the new cluster as long as its weight is lower. Head_ message is limited to transmission within k-hop, so it may happen that some nodes couldn't receive any Head _message. In DSBCA algorithm, if the node does not receive Head_ message in T (w) (T (w) <T (k)), it declares itself the cluster head, where T(w)is waiting time, and T(k)is the refresh time related to distribution of nodes and specific applications. The settings of T (w) and T(k) should ensure that each node in the network can find its own cluster head, and the algorithm restarts the clustering process after T(k)circularly.

3.3. Clusters Building Phase

DSBCA sets the threshold of cluster size. The number of cluster nodes not exceeds the threshold to avoid forming large clusters, which will cause extra overhead and thus reduce network lifetime. When the cluster head node receives Join _message sent by the ordinary node, it will compare the size of cluster with threshold to accept new member and update the count of cluster nodes if the size is smaller than threshold, or reject the request. If the rejected node has cluster head already, the clustering process ceases. Otherwise, it finds another appropriate cluster to join.

Each member node of cluster maintains a clu ster information table, which saves the HID, HD, SID and other information. If a node receives transmitting packet in work, it will update its cluster information table correspondingly. For example, the node checks HD in a newly received packet, if HD is smaller, then it updates the value of HD in table, with SID updated. That is to say, it has found a shorter path to cluster head and sets the new SID as its forwarding node. There is only a single HID entry in the ordinary node because it belongs to one cluster head, but the overlapping cluster node has multiple HID information entries for different clusters. DSBCA algorithm avoids the fixed cluster head scheme (cluster head manages cluster and forwards data, so it consumes energy faster than other nodes), with periodic replacement to balance the node energy consumption.

3.4. CYCLE PHASE

The cluster is stable for a while until the process of reelecting cluster head is triggered in T(k). The cluster head gathers the weight of all member nodes, and then selects the node with highest weight as the next head node. In this way, the communication costsare decreased. The reelecting of cluster head occurs in the 'old' cluster, so the broadcast of temporary head and the corresponding responses of all the k-hops neighbors are unnecessary.

4. SYSTEM MODEL

We first recall some notations and expressions to be used in the main algorithm. A wireless sensor network can be modeled as a graphG=(V, E). Gis a unit disk graph. Vis the set of sensor nodes, andE⊆V 2 is the set of links among the nodes. There is a link between two nodes if they are within

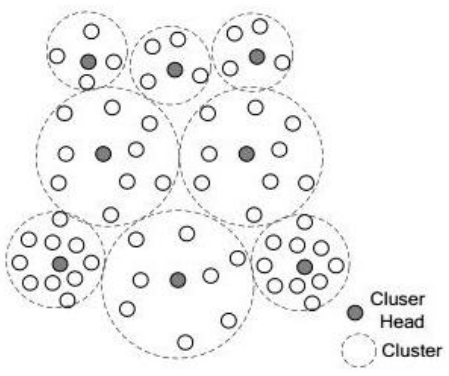
 ISSN: 2321-8169
990– 996

the transmission range of each other. What we discuss are multi-hops WSNs with heterogeneous distribution, and each node is assigned to a unique identifier. The sensor node is not equipped with GPS equipment and couldn't position solely. This paper makes the following assumption. All nodes constitute network topology by selforganizing, and the nodes transmit data with the same signal intensity and with the same maximum distance L in the same frequency (shared channels are freely competitive and faultless). The refreshing time of algorithm is restrained by the switching of nodes state (such as node death). The position of each sensor node(fig 4.1) is fixed in a certain period of time, which is commonly adopted in WSNs modeling. All the sensors are distributed according to a Poisson process with intensity λ. We use the same radio model as stated in [20] where to transmit an l-bit message over a distance d, the power consumption is

$$E_{Tx}(l,d)=E_{Tx-elec}(l)+E_{Tx-amp}(l,d)$$

$$= \begin{cases} lE_{elec}+l\varepsilon_{fsd}\,2\,, & d<d_o \\ lE_{elec}+l\varepsilon_{mpd}\,4\,, & d>d \end{cases}$$

$$E_{Rx}(l)=E_{Rx-elec}(l)=lE_{elec}$$



DSBCA clustering in nonuniform distribution.

Fig 4.1 System Model

The electronics energy Elect depends on the digital coding, modulation, filtering, and reading of the signal, whereas the amplifier energy, εfsd 2 or εmpd 4, depend on the distance to the receiver and the acceptable bit-error rate.

5. PERFORMANCE EVALUATION

In this section, lots of simulation experiments are presented to demonstrate the effectiveness and superiority of the proposed new algorithm in comparison with the previous algorithms. Intuitively, more obvious advantages of DSBCV algorithm.

5.1 THE SIMULATION MODEL

In the simulation experiments, WSNs nodes are randomly distributed in the 50 m×50 m area. We compare DSBCA algorithm with the classical LEACH, HEED and WCA, observing the network lifetime especially. In comparison, we make use of the model and MATLAB 2009b as simulation tool

. In the course of simulation, the target area is set as [0, 50] ×[0, 50], and the base station is in the interval [25, 100]. At the same time, the value of k is fixed as 2, 3 and 4(when k=1, the limited communication distance of nodes may lead to low network coverage because of insufficient

Neighbor nodes; when k>4, excessive big clusters or excessive nodes without cluster may reduce network life cycle), and the threshold is set as 15

To facilitate unified comparison, the survival time is represented by the number of rounds where each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase when the data transfers to the base station. When contrasting algorithms in the same round, the algorithm with lower ratio of dead nodes is deemed better. Similarly, we stipulate that the life cycle ends when 80% nodes are dead in our experiments

The curve of DSBCA is smoother than others, which indicates that the clustering structure is more stable. So DSBCA could adapt to diverse distribution. However, the life cycle of other traditional algorithms drops sharply when the number of nodes exceeds 150. The decline in the fig. 8 is relative to the clustering structure and the selection of cluster head. DSBCA can form more reasonable cluster structure to avoid frequent exchange of the nodes weight information and temporary cluster head broadcasting after the first clustering. As a result, the energy consumption decreases effectively. The cluster structure changes in each round in LEACH, HEED and WCA; nevertheless, DSBCA maintains relatively stable

clustering structure in which switching of cluster head often occurs in the same cluster.

The simulations of clustering structure in 150 nodes of random distribution in the 50 m ×50 m region are shown in

 ISSN: 2321-8169
990– 996

fig. 9–11. The clustering structure is more balanced compared to other algorithms. DSBCA optimizes the energy consumption and ensures the life cycle in non-uniform distribution of adverse circumstances. In fig. 12–14, we can see the simulation results of clustering structure in 200 nodes of random distribution in the 50 m×50 m region, and the same conclusion follows.

6. CONCLUSION AND FUTURE WORK

In this paper, ,we have investigated the performance of our proposal and compared it against competing algorithms through synthesized probabilistic distributions of file chunks. Emerging distributed file systems in production systems strongly depend on a central node for chunk reallocation.

This dependence is clearly inadequate in a large- scale, failure – prone environment because the central load balancer is put under considerable workload that is linearly scaled with the system size, and may thus become the performance bottleneck and the single point of failure.

Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the literature. The simulation results indicate that our proposal is comparable with the existing centralized approach and considerably outperforms the prior distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic.

7. RESULTS

In this section, we see the output of the response time for all the server, the starting time of the server ,the ending time of the server and the response time of the server and which one is the quickest.

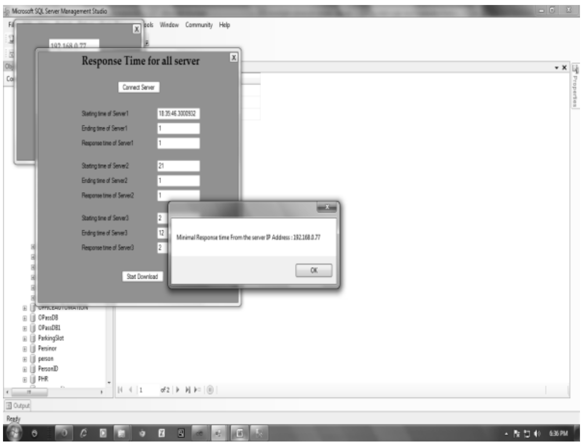


Fig 7.1

The above figure 7.1 shows the server number which one is the quickest response time.

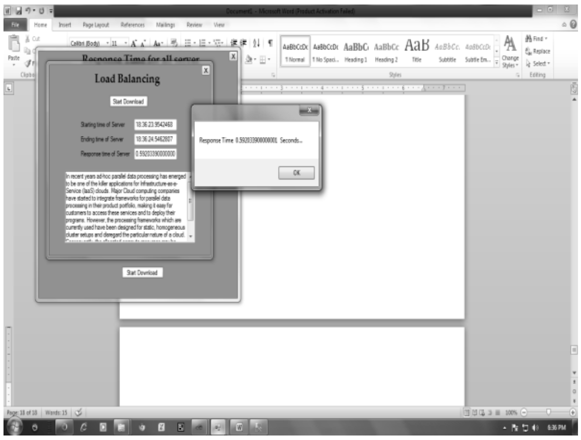


Fig 7.2

In the above figure 7.2 shows the download page it show how much time it take to download the file.

REFERENCES

1. Khesong Huang and Jie Wu, “A Probabilistic Clustering Algorithm in Wireless Sensor Network”, Department of Computer Science and Engineering, Florida Atlantic University.
2. Svahid Hosseini, Abolfazl Toroghi Haghighat and Farshad Eshghi Esfahani, “Designing a Clustering and Prediction-based Protocol for Target Tracking in Wireless Sensor”, Advances in Computer Science: an International Journal, Vol. 2, Issue 3, and No. , 2013.
3. Ying Liao, Huan Qi and Weiqun Li, “Load-Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks”, National Natural Science Foundation march 2013.
4. .I. Stoica, R. Morris, D. Liben-Nowell and H. Balakrishnan, “Chord: A Scalable Peer-to-PeerLookup Protocol for Internet Applications,” IEEE/ACTrans.Networking, vol. 11, no. 1, pp. 17-21, Feb. 2003.
5. S.Acharya, R.Alonso,M.J. Franklin, and S.B. Zdonik, “BroadcastDisks: Data Management for Asymmetric Communications Environments,” Proc.ACM SIGMOD’95, pp.199-210, 1995.
6. D.Barbara´ , “Mobile Computing and Databases: A Survey,”IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 108-117, Jan./Feb. 1999.
7. . N.Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD ’90, pp. 322-331, 1990.

ISSN: 2321-8169
990– 996

8. C. Y. Chow, H. Va Leong, and A. Chan, “Peer-to-Peer Cooperative Caching in Mobile Environment,” Proc. 24th IEEE Int’l Conf. Distributed Computing Systems Workshops (ICDCSW ’04), pp. 528-533, 2004.
9. C.-Y. Chow, H. Va Leong, and A.T.S. Chan, “Distributed Group- Based Cooperative Caching in a Mobile Broadcast Environment,” Mobile Data Management, pp. 97-106, 2005.
10. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, Computational Geometry Algorithms and Applications, second ed. Springer, 2000.
11. T.Hara, “Cooperative Caching by Mobile Clients in Push-Based Information Systems,” Proc. 11th ACM Int’lConf.Information and Knowledge Management, pp.186-193, 2002.
12. . W.-S. Ku, R. Zimmermann, and H. Wang, “Location-Based Spatial Queries with Data Sharing in Wireless Broadcast Environments,” Proc. 23rd IEEE Int’l Conf. Data Eng. (ICDE), 2007.
13. D.C. Montgomery and G.C. Runger, Applied Statistics and Probability for Engineers, second ed. John Wiley & Sons, 1998.

A.Muthulakshmi received her M.SC. degree in Computer Science and Technology (CS&T) in 2010 from Madras University and post graduate degree in Computer Science and Engineering from Anna University Chennai, India. Her areas of interest are Mobile computing, operating systems and Computer Networks. she has presented many papers in national conferences in various fields. As part of this paper, she is working on developing load-balanced clustering algorithm for WSNs that can support file and database access. She is also investigating operating systems support for mobile hosts. she is a member of ISTE.