Distributed Algorithm to Load Balanced Data Aggregation Tree Problem under Deterministic Network Model

Shaik Rizwana Begum
CSE Dept., JNTUA College of Engineering, Anantapuramu, A.P, India.

Dr. S. Vasundra
Professor and H.O.D, CSE Dept., JNTUA College of Engineering, Anantapuramu, A.P, India.

Abstract – In Wireless Sensor Networks (WSNs) data gathering is a fundamental task. Data gathering trees have the ability of performing aggregation operations are also known as Data Aggregation Trees (DATs). Load-balanced factor is neglected in construction of DATs in previous work. By investigating the problem, load-balanced DAT are constructed in which load-balanced factor is considered in three phases, Load–balanced Maximal Dominating set, Connected Maximal Dominating Set, Load-balanced Parent Node Allocation. But these three phases may lead to either performance loss or improvement, since it has not been investigating correlations among them. Therefore, in this paper, we propose a sophisticated model to integrate the above three phases together and analyze the overall performance of LBDAT construction, by designing a distributed algorithm. Simulation results show that proposed algorithm better than the existing approaches significantly.

Index Terms – Load-balanced, Deterministic Network, Load-Balanced Maximal Dominating Set, Connected Maximal Dominating Set, Distributed algorithm.

1. INTRODUCTION

Wireless sensor networks (WSNs) refer to a group of spatially dispersed and dedicated sensor nodes [13], which sense the monitored environment periodically and send the information to the base station. At the base station gathered or collected information can be processed further for end-user queries. In this data gathering process, data aggregation can be used to combine data from different sensors to eliminate redundant transmissions, as the data sensed by different sensors have spatial and temporal correlations [1], [2]. By using this data aggregation technique in WSNs, the amount of data to be transmitted by a sensor is reduced. This in turn cause decrease in each sensor's energy consumption so that whole network life time is extended.

A tree based topology is adopted for continues monitoring applications to gather and aggregate sensing data [3]. Data gathering trees have the ability of performing aggregation operations are also known as Data Aggregation Trees (DATs). DATs are directed trees rooted at base station and have a unique path from each node to the base node. But the construction of DATs did not consider load-balance factor, due to this nodes may quickly lose their energy.

By investigating the above problem DAT is constructed by considering load-balance factor, then the DAT is referred as LBDAT (load-balanced DAT). LBDAT is constructed under DNM in three phases LBMDS, CMDS, LBPNA. As these three phases algorithms may lead to performance loss or improvement, since it has not been investigating correlations among them.

In this paper, we propose a sophisticated model to integrate the above three phases together and analyze the overall performance of LBDAT construction, by designing a distributed algorithm.

2. RELATED WORK

A lot of research work has been done on aggregation trees in wireless sensor networks. There were many changes applied to the aggregation trees for maintain the tree in wireless sensor network. Some of those efforts are shown below.

- DATs
- Load Balanced DATS

2.1. Data aggregation Trees (DATs)

DATs which will perform aggregation operations on data gathering trees are directed trees rooted at the base station and have single directed path from each node to the base station [4]. In DAT, sensing data are combined at intermediate sensors which are collected from different sensors; to certain aggregation operations include COUNT, SUM, MAX AVERAGE and MIN [5].

Most of the existing DAT constructions are based on the Deterministic Network Model (DNM), where any pair of nodes in a WSN is either disconnected or connected [6]. In this model, any pair of nodes is neighbors if their physical distance is lesser than the communication range and the rest of pairs are always disconnected.

In the construction of DAT load balance factor has not been considered. If we didn't consider balancing the traffic load among the nodes in a DAT, some heavy-loaded nodes may quickly lose their energy, which in turn might cause network partitions or malfunctions.

2.2. Load-Balanced DAT (LBDAT)

DAT construction problem is the major concern, whereas the previous works focused on the aggregation problem and didn't consider balancing traffic load among all the nodes in a network. Load- Balanced DAT considers the load-balance of the network [7]. Construction of LBDAT under the DNM is in three phases:

- Load Balanced Maximal Dominating Set (LBMDS) An Maximal Dominating Set define formally [8] as consider a graph $G = (V, E)$, an Dominating Set (DS) is a subset $D \subseteq V$ such that for any two vertex $v_1, v_2 \in I$ they are not adjacent, i.e., $(v_1, v_2) \notin E$. A DS is called an MDS if we add one more arbitrary node to this subset, the new subset will not be a DS any more.
- Connected Maximal Dominating Set (CMDS)After getting an LBMDS, we have to find a minimum-sized set of nodes called LBMDS connector set C to make this LBMDS M connected, which is a Connected MDS [9].
- Load Balanced Parent Node Allocation (LBPNA) after obtaining a CMDS, we have to find a parent for the connected MDS nodes, which is called LBPNA [10]. When LBPNA is decided, by assigning a direction to each link in the tree, we obtain an LBDAT.

These Algorithms can extend network lifetime significantly, but lead to either performance loss or improvement, as the correlations among them have not been investigated.

3. PROPOSED WORK

In this paper, we propose a sophisticated model to integrate the above three phases together and analyze the overall performance of the LBDAT construction problem. To solve the LBDAT construction problem under DNM, distributed algorithm is designed which will integrate all the above three phases.

3.1. p-norm

The load-balanced factor is our major concern of this work. Thus, finding a suitable dimension to evaluate load-balance is the key to solve the CMDS and LBPNA problems. We use p-norm to calculate load-balanced in this paper. p-norm can be given as follows:

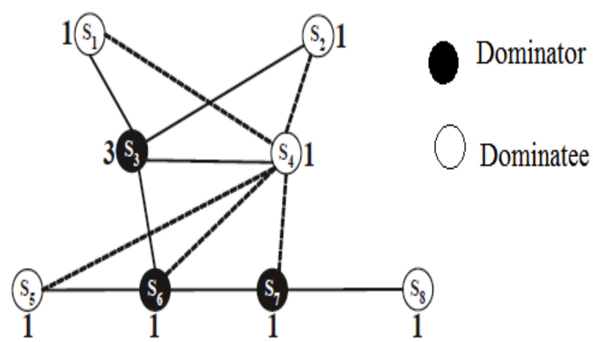
The p-norm of a $n \times 1$ vector $X = (x_1, x_2, x_3 \ldots\ldots, x_n)$ is:

$$|X|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} \quad (1)$$

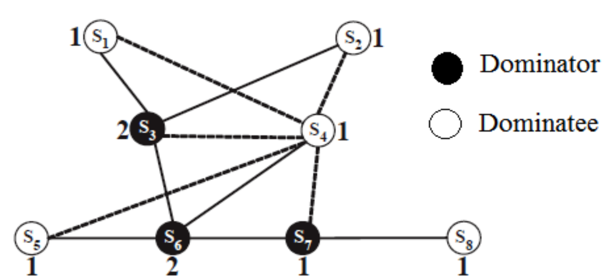
P-norm shows different properties for different values of p stated in [11]. If p is near to 1, the information routes are similar to the geometric shortest paths from the source to the base stations. For p=2, the information flow shows a similarity to electrostatics field. This can be used to calculate the load-balance among x_i. If the p-norm value is smaller, the more load-balanced is concerned feature vector X.

3.2. Valid Degree (VD)

For each dominate s_i, VD_i is the number of its connected dominators. For eeach dominator s_j, VD_j is the number of its allocated dominates (Dominating Set (DS) is a subset of nodes in the network where every node is either in the subset or a neighbor of at least one node in the subset. To be useful DS should be connected namely Connected Dominating Set (CDS). The nodes in CDS are called dominators, otherwise, dominates. In a WSN with CDS, dominates forward their data to their connected dominators. The CDS with the smallest size is called Minimum-sized Connected Dominating Set (MCDS)).



(a)



(b)

Fig. 1 Allocation examples

Fig. 1(a) and Fig. 1(b) give imbalanced and balanced allocations of dominates. Using VD_i as information vector X, and can use p-norm to measure the load-balanced of the dominate allocation scheme. Therefore, the p-norm value of the allocation scheme shown in Fig. 1(a) is $\sqrt{6}$ and Fig 1(b) p-norm value is $\sqrt{6}$. $\sqrt{6}$ <$\sqrt{8}$, which says that allocation shown in Fig. 1(b) is further load-balanced than the Fig. 1(b).

Because of instability of network topology, it is not always practical to allocate one dominate to one dominator. As to adapt network topology change, a terminology Expected Allocation Probability (EAP) is used as follows:

3.3. Expected Allocation Probability (EAP)

For each dominatee and dominator pair, there is an EAP, which gives the predictable probability so that the dominatee is to be allocated to the dominator.

The EAP value associated on each dominatee and dominator pair directly determines the load-balanced factor of each allocation. Then the properties of the EAP values as follows:

1. For each dominate $s_i, \sum_{j=1}^{|NE(s_i)|} EAP_{ij} = 1.$ where $NE(s_i)$is the set of neighboring dominators of s_i, $|NE(s_i)|$ is the number of the nodes in set $NE(s_i)$;
2. In order to produce the most load-balanced allocation, which is obtained when the expected number of allocated dominated of all the dominators are the same. It can be formulated as follows:

$$EAP_{i1} \times VD_i = \cdots = EAP_{i|NE(s_i)|} \times VD_{|NE(s_i)|} \quad (2)$$

3.4. Distributed Algorithm

The objective of the LBDAT problem is to find a load-balanced dominate allocation scheme. The most load-balanced allocation is the expected number of allocated dominatees of all the dominators is same, which is formulated n equation 2. By listing all the equations, we can solve them to get EAP_{ij} of each connected dominatee s_i, which is given as follows:

$$EAP_{i1}: EAP_{i2}: \ldots\ldots EAP_{i|ND(s_i)|} = VD_2 \times VD_3 \times \ldots . VD_{|ND(s_i)|}: \ldots\ldots.: \prod_{j=i,i\neq j}^{|ND(s_i)|} VD_j : ..: VD_1 \times VD_2 \times \ldots VD_{|ND(s_i)|-1} \quad (3)$$

Therefore, the distributed LBDAT problem can be transformed to calculate the EAP value of each dominatee locally.

The distributed algorithm is a localized two-phase algorithm where every node only needs to know the connectivity data in its 1-hop neighborhood. All the nodes get the VD values by broadcasting messages to all its neighbor nodes, and then store the values locally. Each dominate calculates the EAP values using equation 3.

We call the following algorithm as LBDAT-Distributed algorithm. We use the following terms in algorithm,

VD_k: The VD value of each node s_k.

$ND(s_k)$: The set of neighbor dominatees of dominator s_k.

$|ND(s_k)|$: The number of nodes in set $ND(s_k)$.

$NE(s_k)$: The set of neighbor dominators of dominatee s_k.

$|NE(s_k)|$: The number of nodes in set $NE(s_k)$.

EAP_{i1}: The EAP value of each connected dominatee s_i and dominators_j pair.

Algorithm: Distributed LBDAT

1: Initialization Phase:

2: For each dominatee s_i, get the number of neighbor dominators ($|NE(s_i)|$) and store locally.

3: For each dominator s_j, get the number of neighbor dominates ($|ND(s_j)|$) and store locally.

4: **Allocation Phase:**

5: For each dominate$e$ s_i, calculate its neighboring dominators EAP_{ij} by the following formula:

6: EAP_{i1}: : EAP_{i2}: ...: $EAP_{i\,|ND(s_i)|} = VD_2 \times VD_3 \times \ldots \times VD_{\,|ND(s_i)|}: \ldots\ldots: VD_1 \times VD_2 \times \ldots . \times VD_{|ND(s_i)|-1} = \prod_{j\neq 1,i\neq j}^{|ND(s_i)|} VD_j$

Each node s_i maintains the following data structures:

1) s_i's ID, initialized to 0.
2) The dominator/dominatee flag f. 1 means dominatee. It is initialized to 0.
3) $|ND(s_i)|$, if s_i is a dominator; $|NE(s_i)|$, if s_i is a dominatee, initialized to 0.
4) Neighbor dominator/dominatee lists. A list contains: a dominator/ dominatee's ID, its VD value, and EAP_{ij}, initialized to$\emptyset$.

Initially, each node initializes its data structures and a broadcast a hello message containing its ID, VD, and f to its 1-hop neighbor to exchange neighbor's information. All the nodes run the following:

For any dominator s_i, upon receiving a hello message from node s_j: if s_j is a dominator, ignore the message. If s_j is a dominatee, update $|ND(s_i)|$ and dominatee $s_j's$ ID and VD value in the neighbor dominatee list of the dominator f s_i.

For any dominator f s_i, upon receiving a hello message from node f s_j: if f s_j is a dominatee, ignore the message. If f s_j is a dominator, update $|NE(s_k)|$: and dominator $s_j's$ ID and VD value in the neighbor dominator list of the dominatee s_i. Calculate and store EAP_{ij} based on the VD values stored in the neighbor dominator list using equation 3.

The distributed algorithm is a 2-phase algorithm. The first phase is the initialization phase, where all the nodes get its neighborhood information and update its own data structure locally. In practical, it is hard to decide when the initialization phase completes. Hence we set a timer. If the timer expires, the second phase, allocation phase, starts to work. In the allocation phase, every dominatee calculates the EAP values of its connected dominators using equation3. We only use 1-hop neighborhood information to calculate the EAP values locally. Therefore, it is an easy and efficient algorithm. Nevertheless, only using the 1-hop neighborhood information to calculate the EAP values may lead us to find a local optimal solution instead of a global optimal solution.

4. SIMULATION

Here we evaluate our proposed algorithm by comparing it with pervious work [12], in which each dominatee chooses the neighbor dominator of the smallest ID as its parent.

Four schemes are implemented:

1. LBCDSs with LBDAT noted as LB-A
2. LBCDSs with smallest ID dominator, noted as LB-ID
3. MDS-based CDS with LBDAT, noted as MIS-A
4. MDS-based CDS with the smallest ID dominator selection, noted as MIS-ID work in [6].

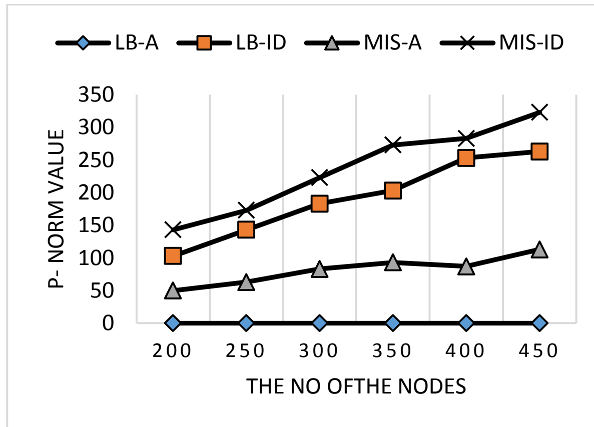
We compare them in terms of the p-norm values.

4.1. Simulation Environment

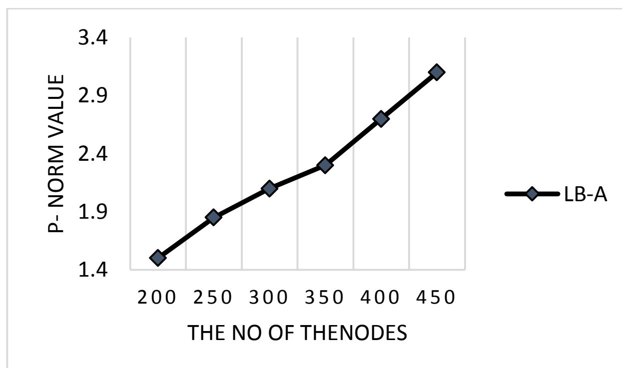
We build our own simulator; here all nodes have the same communication range (10m). n nodes are randomly deployed in a fixed area of $100m \times 100m$. n is incremented from 200 to 450 by 100. For a certain n, 100 instances are generated. The results are average among 100 instances.

4.2. Simulation Results

Fig 2 shows the p-norm values of the four schemes. With the increase of the number of the sensor nodes, the p-norm values increase correspondingly. This is because when the number of the nodes increases, we need more nodes to build an LBCDS. According to p-norm values, lesser the p-norm value, more the load-balanced the scheme. Fig 2 MDS-based CDS scheme has the largest p-norm values while the LBCDS with LBDAT scheme has the smallest p-norm values. This because the MDS-based CDS scheme did not consider the load-balanced factor when building a CDS and then allocating dominatees to dominator. For clearly to say the p-norm values of the LBDAT scheme, we redraw the curve using smaller scale in Fig 2(b) for LBCDS with LBDAT. Fig 2 demonstrates that the Distributed LBDAT algorithm fits for any type of CDS.



(a)



(b)

Fig 2 p-Norm Value

5. CONCLUSION

In this paper, we propose a new Distributed LBDAT concept, which is a CDS with minimum p-norm value in order to assure that the work load among each dominator is balanced. And aims to load-balanced allocate each dominatee to a dominator. We use EAP value to represent the expected probability of the allocation between each dominatee and dominator pair. The extensive simulation results show that compared to the existing, using an LBCDS and EAP values to load-balanced allocate dominatees can prolong network lifetime significantly.

REFERENCES

[1] G. Hadim and N. Mohamed, ''Middleware Challenges and Approaches for Wireless Sensor Networks,'' IEEE Distrib. Syst. Online, vol. 7, no. 3, pp. 1-23, Mar. 2006.
[2] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, ''On Network Correlated Data Gathering,'' in Proc. IEEE INFOCOM, 2004, pp. 2571-2582.
[3] S. Cheng, J. Li, and Z. Cai, ''Oð"Þ-Approximation to Physical World by Sensor Networks,'' in Proc. IEEE INFOCOM, 2013,pp. 3084-3092.
[4] H.O. Tan and I. Korpeogle, ''Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks,'' SIGMOD Rec., vol. 32, no. 3, pp. 66-71, Dec. 2003.
[5] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler, ''Supporting Aggregate Queries over Ad-HocWireless Sensor Networks,'' in Proc. IEEE WMCSA, 2012, pp. 49-58.
[6] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin,''Statistical Model of Lossy Links in Wireless Sensor Networks,''in Proc. IPSN, 2005, pp. 81-88.
[7] Jing He, ShoulingJi, "Constructing Load-Balanced Data Aggregation Trees in Probabilistic Wireless Sensor Networks", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 7, JULY 2014, pp.1681-1690.
[8] P. Raghavan and C.D. Thompson, ''Randomized Rounding: ATechnique for Provably Good Algorithms and Algorithmic Proofs,'' Combinatorica, vol. 7, no. 4, pp. 365-374, 1987.
[9] P.J. Wan, S.C.-H. Huang, L. Wang, Z. Wan, and X. Jia, ''Minimumlatency Aggregation Scheduling in Multihop Wireless Networks,'' in Proc. MobiHoc, 2009, pp. 185-194.
[10] Y. Wu, S. Fahmy, and N.B. Shroff, ''On the Construction of a Maximum-Lifetime Data Gathering Tree in Sensor Networks: NP-Completeness and Approximation Algorithm,'' in Proc. IEEE INFOCOM, 2008, pp. 356-360.\
[11] M. Kalantari, M. Haghpanahi, and M. Shayman, A p-norm Flow Optimization Problem in Dense Wireless Sensor Networks, INFOCOM'08, Phoenix, AZ, USA, April 13-18, 2008.
[12] P.J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks", MobiHoc 2009, New Orleans, Louisiana, May 18-21, pp. 185-194, 2009.
[13] Sercan VANÇİN, Ebubekir ERDEM, "Design and Simulation of Wireless Sensor Network Topologies Using the ZigBee Standard", International Journal of Computer Networks and Applications (IJCNA), Volume 2, Issue 3, May – June (2015).

Authors

Shaik Rizwana Begumreceived B.Tech degree in Computer Science and Engineering from Safa College of Engineering and Technology Kurnool, affiliated to JNTUA University, Anantapuramu, A.P, India, during 2008 to 2012. Currently pursuing M.Tech in Computer Science from JNTUA College of Engineering, Anantapuramu, A.P, India, during 2013 to 2015 batch. Her Area of interests include Computer Networks, Network Security.

Dr S. VASUNDRA, presently working as Professor and Head of the Department CSE, JNTUA CEA. She completed her Ph.D from JNTUA University, anantapur, M.Tech from JNTUA and B.E from VTU. She is having 16 years of teaching experience and 5 years of research experience. Published 20 papers in various international journals and 3 in national journals. Her areas of interest include MANET's, Cloud Computing, Algorithms, Data Structures and Distributed Computing.