Distributed Algorithms to Form Cluster based Spanning Trees in Wireless Sensor Networks

Kayhan Erciyes[1], Deniz Ozsoyeller[2], and Orhan Dagdeviren[3]

[1] Ege University International Computer Institute
Bornova, Izmir, TR-35100, Turkey
kayhan.erciyes@ege.edu.tr
[2] Izmir University of Economics, Computer Eng. Dept.
Balcova, Izmir TR-35350, Turkey
deniz.ozsoyeller@ieu.edu.tr
[3] Izmir Institute of Technology Computer Eng. Dept.
Urla, Izmir TR-35340, Turkey
orhandagdeviren@iyte.edu.tr

Abstract. We propose two algorithms to form spanning trees in sensor networks. The first algorithm forms hierarchical clusters of spanning trees with a given root, the *sink*. All of the nodes in the sensor network are then classified iteratively as *subroot*, *intermediate* or *leaf* nodes. At the end of this phase, the local spanning trees are formed, each having a unique subroot (clusterhead) node. The communication and data aggregation towards the sink by an ordinary node then is accomplished by sending data to the local subroot which routes data towards the sink. A modified version of the first algorithm is also provided which ensures that the obtained tree is a breadth-first search tree where a node can modify its parent to yield shorter distances to the root. Once the sub-spanning trees in the clusters are formed, a communication architecture such as a ring can be formed among the subroots. This hybrid architecture which provides co-existing spanning trees within clusters yields the necessary foundation for a two-level communication protocol in a sensor network as well as providing a structure for a higher level abstraction such as the γ synchronizer where communication between the clusters is performed using the ring similar to an α synchronizer and the intra cluster communication is accomplished using the sub-spanning trees as in the β synchronizers. We discuss the model along with the algorithms, compare them and comment on their performances.

Key words: spanning tree, clustering, synchronizers, wireless sensor networks

1 Introduction

Wireless Sensor Networks (WSNs) have important scientific, environmental, medical and military applications. Example WSN applications include habitat monitoring, remote patient monitoring and military defense systems [1]. WSNs may consist of hundreds or even thousands of nodes that operate independently. A survey of WSNs can be found in [2]. WSN nodes are small, inexpensive, embedded, require low power and are distributed regularly or irregularly over a significantly large area. WSN nodes are usually deployed in highly dynamic and sometimes hostile environments. It is therefore very important that these networks should have the capability to perform unattended and distributed but coordinated operation with the other nodes and also to provide self-healing in the case of faults.

Communication in WSNs can be performed using two fundamental approaches as tree based and cluster based. Cluster based communications require grouping of closely coupled elements of the sensor network into clusters and electing one of these nodes as the clusterhead (cluster leader) [3]. The cluster leader provides the coordination of the communication among the cluster members and other clusters. Energy is an important and crucial resource in sensor networks due to the limited lifetime of sensor batteries and also difficulty of recharging batteries of thousands of sensors in remote or hostile environments. Communication of sensor nodes dominate their energy consumption even when they are at idle-listening state [4].

In this study, we propose a distributed algorithm that forms hierarchical spanning trees in a WSN where each sub-spanning tree has a root node that has the role of the leader for that subtree. Our algorithm has the topology of a spanning tree but also has a cluster structure with a clusterhead, therefore is an integration of the tree based and cluster based approaches. To our knowledge, the algorithm in this study is the first attempt to provide a hybrid approach for communication in WSNs. The rest of the paper is organized as follows. Section 2 provides the related work and the algorithms designed are detailed in sections 3 and 4 along with analysis and results obtained. Finally, conclusions are presented in Section 5.

2 Background

2.1 Clustering in WSNs

A WSN can be modelled by a graph $G(V, E)$ where V is the set of vertices (nodes of WSN) and E is the set of edges (communication links among the nodes). Clustering the nodes of a graph or *graph partitioning* is *NP-Hard*. For this reason, clustering in WSNs is usually performed using some heuristics. Some of the benefits to be gained from clustering in mobile ad hoc and WSNs are the reduction in energy for message transfers and forming of a virtual backbone for routing purposes [5].

HEED (Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach) [6], proposes a distributed clustering algorithm for sensor networks. Clusterheads in HEED are elected using a probabilistic heuristic that considers the residual energy of a node and the number of its neighbors (its degree). HEED assumes a homogenous network and also that neighbor connectivity is known and provides balanced clusters. LEACH (Low-Energy Adaptive Clustering Hierarchy) [7] provides rotating clusterheads chosen randomly and assumes clusterheads consume uniform energy. Both HEED and LEACH find clusters in a finite number of steps. In PEAS [8], a node goes to sleep (turn off its radio) when it detects a routing node in its transmission range.

In GAF [9], the sensor network is divided into fixed square grids each with a routing node. Communication to the sink is propagated by the routers. The ordinary nodes in each grid can turn off their radio components when they have no transmission. GEAR (Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks) [10] and TTDD (Two-tier Data Dissemination Model for Large Scale Wireless Sensor Network) [11] are examples of other protocols for cluster formation in WSNs.

2.2 Spanning Tree Formation in WSNs

Building spanning trees rooted at a sink node for data collection is a fundamental method for data aggregation in sensor networks. However, due to the nature of the sensor networks, the spanning tree should be formed in a decentralized way. Gallagher, Humblet and Spira [12], Awerbuch [13], Banerjee and Khuller [14] have all proposed distributed spanning tree algorithms. Gallagher, Humblet and Spira distributed algorithm determines a minimum weight spanning tree for an undirected graph by combining small fragments into larger fragments. A fragment of a spanning tree is its subtree. Time complexity for this algorithm is O(NlogN).

The ENCAST (ENergy Critical node Aware Spanning Tree) algorithm [15] finds a shortest path tree (SPT) by breadth first traversal from the sink, and each node can reach the sink via the minimum number of hops using this SPT. However, there may be more than one SPTs in dense sensor networks due to the fact that nodes have many neighbor nodes and some of these neighbors have the same minimum-hop distance from the sink energy of a node as the second selection criteria and attempts to label nodes with less energy as leaf nodes.

3 The Distributed Spanning Tree Algorithm

The first algorithm is a modification of the distributed spanning tree formation algorithm for general networks. We modify this general algorithm below so that clusters which are subtrees are also formed with energy considerations of the WSN. We assume that the sensor nodes are distributed randomly and densely over the area to be monitored and the sensor field can be mapped into a two dimensional space. Furthermore, all the sensor nodes have identical and fixed transmission ranges and hardware configurations and each sensor node can monitor its power level E_P.

3.1 Description of the Algorithm

The algorithm we propose is described informally as follows. The sink periodically starts the algorithm by sending a $PARENT$ message to its neighbors. Any node i that has not received a $PARENT$ message before sets the sender as its parent, sends $ACK(i)$ message to its parent and sends a $PARENT(i)$ message to all of its neighbors. We provide a *depth of subtree* parameter d as the modification to the above classical algorithm to form a spanning tree. Every node that is designated a parent performs *n_hops = (n_hops + 1) MOD d* to append to its outgoing message. The recipient of the message with *n_hops =0* are the *SUBROOT*s, and *n_hops ¡=d* are *INTERMEDIATE* nodes or *leaf* depending on their level within a subtree.

The state diagram of Fig. 1 depicts the operation of the Distributed Spanning Tree Algorithm (DSTA). The algorithm is initiated by the sink at regular intervals. Any ordinary node that has not been labeled before, receiving a PARENT message from an upper node, labels itself according to the number of hops the message has traveled which is shown by the parameter of the PARENT message.

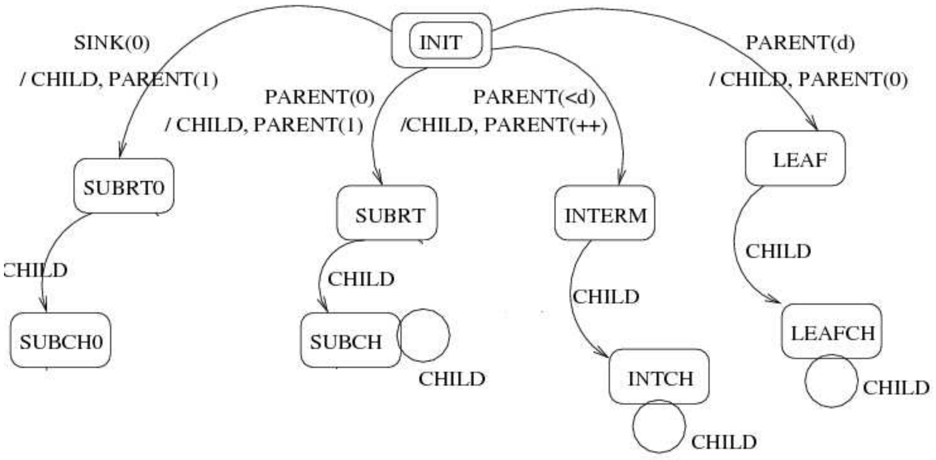

Fig. 1. The Finite State Machine Diagram of DSTA.

Any further change of states between subroot, intermediate and leaf nodes are not shown for simplicity. The following is a list of messages used in DSTA :

- $PARENT$: Sent by a parent to the neighbors soliciting for children.
- $CHILD$: Sent by the child to parent acknowledging to be a successor.
- $TIMEOUT$: Internal message informing a timeout has ocurred. This message prevents a subroot waiting indefinitely for acknowledgements from potential children.

The message contains the following fields :

- *Sender* : SINK, SUBROOT, SUBROOT0, INTERMED, LEAF;
- *type* : PARENT, CHILD;
- *n_hops*: integer showing the number of hops the message has travelled.

If the number of hops in the message is equal to zero, the node labels itself as the $SUBROOT$. Else if the number of hops is smaller than the allowed depth d of the sub-tree, the node is an intermediate ($INTERM$) node. Once the number of hops equals the depth, the node is classified as a $LEAF$. Each labeled node acknowledges its parent by the $CHILD$ message. The following is the list of sensor node states:

- $SUBRT$: A node is labeled as a subroot as the message it has received from its parent has n_hops = 0.
- $SUBCH$: A Subroot node has at least one confirmed child in the local tree.
- $INTERM$: A node is an intermediate node, that is, it is not a subroot or a leaf node.
- $INTCH$: An intermediate node with at least one child
- $LEAF$: A node that is the leaf of a local spanning tree.
- $LEAFCH$: A leaf node with at least one child.
- $SUBRT0$: A subroot node that has received a SINK message
- $SUBCH0$: A subroot 0 node that has at least one child.

Remark 1. Energy Considerations : A sensor node rejects being labeled as subroot if its energy level is below a threshold, for example, two thirds of E_P. This is required as a subroot will have more message transfers than an ordinary node.

A branch of the spanning tree formed constitutes a cluster where a subroot node is the clusterhead. Subroots may have other attributed roles in application specific settings. For our purpose, each subroot has the capability to manipulate or filter any incoming message to it during convergecast.

3.2 Analysis of DSTA

In this section, we analyze the number of communication steps (count of messages) to form the spanning trees using DSTA and comment on its performance. Based on the state machine of Fig. 1, the labeling of a sensor node as SUBROOT, INTERMED or LEAF requires two messages called $PARENT$ and $CHILD$. The first message is sent by the parent soliciting children and the second mesage is the acknowledgement of the child to its parent.

Theorem 1. *Time complexity of DSTA is $O(D)$ where D is the diameter of the network from the sink to the furthest leaf and its message complexity is $O(n)$.*

Proof. The time required for the algorithm is clearly the diameter D of the network. Once a node is labeled and has a designated parent, it will only send a message to its neighbors once. If Δ is the maximum degree of the network graph, total number of messages is Δ*n and for small Δ, message complexity is $O(n)$.

3.3 Results

The distributed spanning tree algorithm is implemented with the $ns2$ simulator. The IEEE 802.11g standards are chosen for lower layer protocols. Total number of nodes vary from 100 to 500 nodes. Different size of flat surfaces are chosen for each simulation to create high dense, dense and medium connected topologies to measure the effect of node degree. Surface areas vary from 2700m $\times$ 1200m to 17920m $\times$ 1920m. Depth parameter is changed to obtain different number of clusters, as well as, *SUBROOT*, *INTERMEDIATE* and *LEAF* nodes.

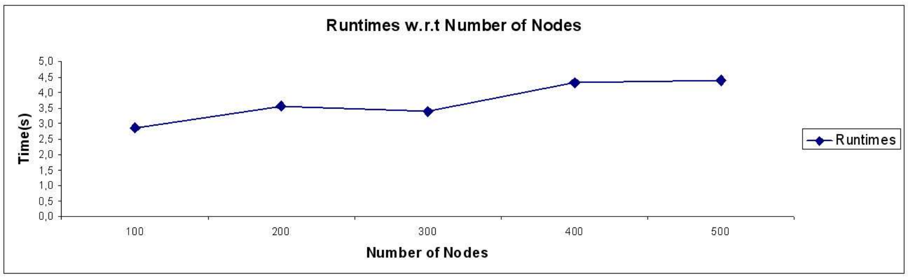

Fig. 2. DSTA Run-times against the Number of Nodes.

Fig. 2 displays the run-time results of the distributed spanning tree algorithm ranging from 100 to 500 nodes. Run-time values increase almost linearly, except for the case of 300 nodes which may be due to their random distribution, indicating the scalability when the total number of nodes is increased from 100 to 500 nodes. 4.5s is needed for the formation of distributed spanning tree with clusters. For a network with 100 nodes, different topologies are created to measure the effect of the average node degree parameter. Because each node must be informed by its neighbors to complete reliable flooding. Any corrupted message must be retransmitted. Fig. 3 shows that algorithm performs well up to high dense topologies with 8 nodes connected on the average.

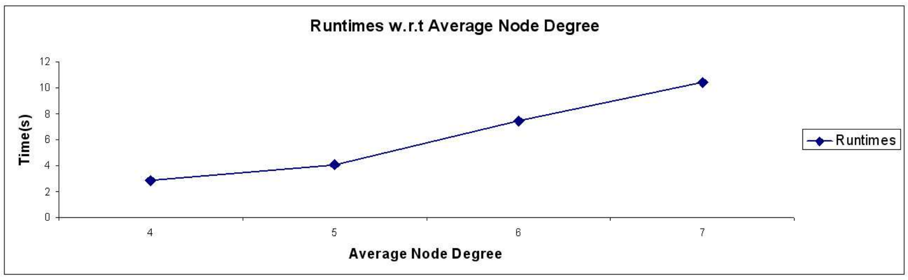

Fig. 3. DSTA Run-times against the Average Node Degree.

Depth parameter of the DSTA changes the number of clusters and the node states in WSN. Number of *SUBROOT*, *INTERMEDIATE* and *LEAF* nodes are measured for 300 nodes as shown in Fig. 4. As depth parameter is increased from 2 to 6, *SUBROOT* node count decreases and *INTERMEDIATE* count increases as expected. Number of *LEAF*s, which are mostly the gateway nodes, decreases same as *SUBROOT*s with depth parameter.

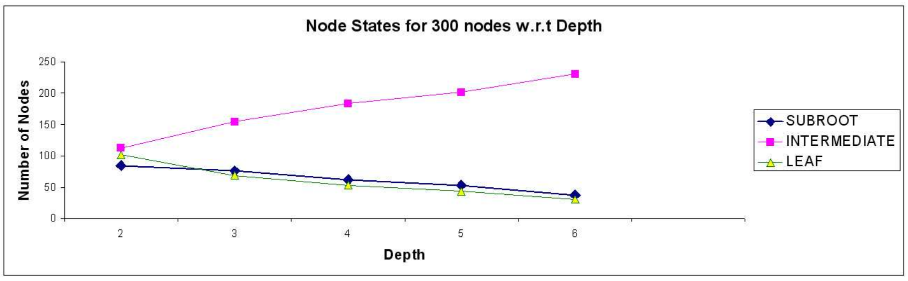

Fig. 4. DSTA Number of Node States against the Depth.

Our results conform with the analysis that the run-time values and message counts grow linearly. Also, algorithm is stable under different node degrees. Depth parameter changes the number of clusters and node states and its selection is very important. Worst delivery times are scalable and show that nodes can route their packets on top of this spanning tree with reasonable delays.

4 Breadth-First Search based DSTA

The second algorithm we propose for spanning tree formation in WSNs is the modification of the Breadth-First Search (BFS) spanning tree algorithm for general networks shown below :

1. Initially, the root sets L(root) = 0 and all other vertices set L(v) = ∞.
2. The root sends out the message Layer(0) to all its neighbors.
3. A vertex v, which gets a Layer(d) message from a neighbor w checks if d + 1 < L(v). If so, it does the following:
   - parent(v) = w;
   - L(v) = d + 1;
   - Send Layer(d + 1) to all neighbors except w.

We apply the algorithm above, however, based on their designated distances, nodes are labeled as*ROOT*, *SUBROOT* and *LEAF* as in DSTA.

Theorem 2. *Time complexity of BFS-DSTA is* O(n) *and its message complexity is* O(n‖E‖).

Proof. As the longest path in a network has n-1 nodes, time complexity of the general asynchronous BFS spanning tree algorithm is *O(n)*. Since at every step, there will be a maximum of $|E|$ messages, the message complexity is *O(n—E—*. For BFS-DSTA, general rules apply and the complexities are the same as the asynchronous BFS algorithm

4.1 Results for BFS based DSTA

We implemented BFS-MDSTA (DSTA with multiple sinks using BFS) in a similar setting of $ns2$ as in DSTA. Fig. 5 shows the running times of BFS-MDSTA for 1,3 and 5 sinks. We see that there is a linear increase as the number of nodes are increased and also running times depend linearly on the number of concurrent sinks.

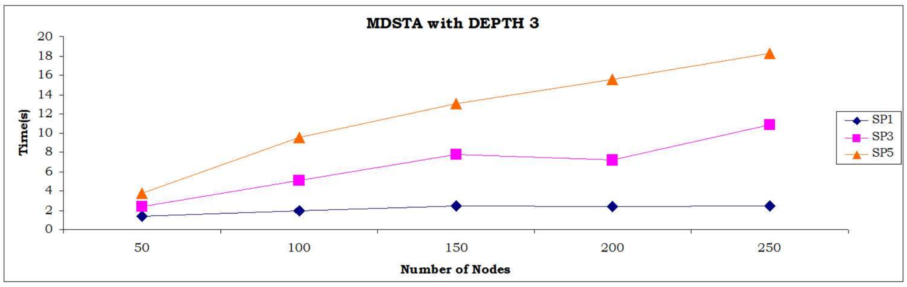

Fig. 5. The Running Times for Multi-sink Formation with BFS-MDSTA.

Fig. 6 shows the number of clusters formed when BFS-MDSTA is applied to a WSN for 1,3 and 5 sinks with a constant depth of 3. The curves are almost identical showing an even distribution of clusters independent of the count and location of the sinks.

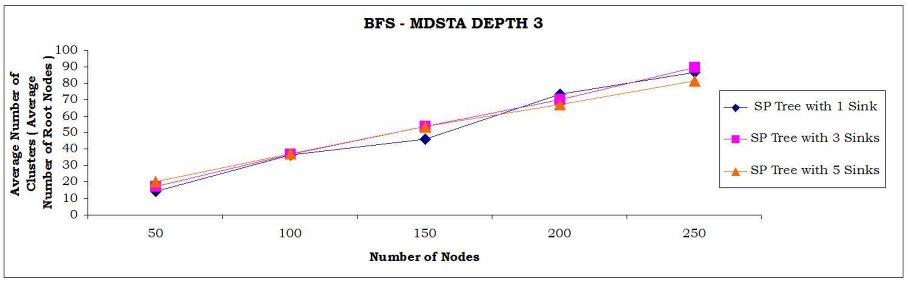

Fig. 6. The Average Number of Clusters for BFS-MDSTA for $d = 3$.

Fig. 7 shows the effect of the subtree depth d on the cluster count when BFS-DSTA is applied upto 250 WSN nodes which was the upper limit that the simulator could tolerate due to the data complexity of maintaining 5 concurrent sinks. We see here that the count of clusters decrease linearly as d increases which is expected.

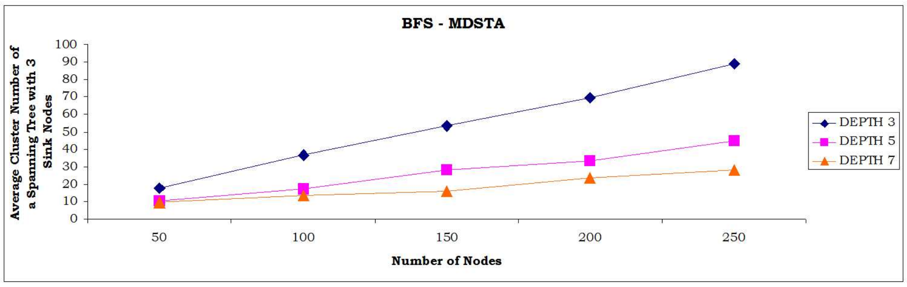

Fig. 7. The Average Number of Clusters for BFS-MDSTA against Depth.

5 Discussions and Conclusions

We proposed two distributed algorithms for spanning tree formation to provide a communication infrastructure in sensor networks. The first algorithm (DSTA) has a lower message complexity of $O(n)$ but does not necessarily find the shortest route to the sink. The second algorithm (BFS-DSTA) uses BFS property and finds the shortest route with elevated message complexity of $O(n|E|)$. These algorithms may be activated at regular intervals by the sink and the dynamic spanning tree configuration consisting of healthy nodes only discards the sensor nodes that have ceased functioning due to energy loss or other hostile environment conditions. We showed that these algorithms are scalable and provide balanced clusters which consist of tress within the clusters. This architecture may be suitably used for a γ synchronizer which requires the same structure we propose. One future direction of this work would therefore be another communication structure such as a ring between the clusterheads so that an α synchronizer can be constructed among the clusters.

The local spanning trees produced by DSTA and BFS-DSTA naturally comprise clusters of the sensor network and therefore can be used for other resource management tasks in sensor networks other than the communication infrastructure or the synchronizer function described in this study. The subroot nodes are the leaders of the clusters that can act as the representatives of their cluster members for various tasks in the sensor networks. These leaders can be connected in various configurations such as the ring or other in order to perform tasks such as mutual exclusion in sensor networks. Advantage of this hybrid approach would be the simple and fast data aggregation using the spanning tree within the cluster and a more general framework such as ring based communication among the clusters. Our work is ongoing and we are looking into labeling some nodes of the WSN as privileged nodes of improved transmission capabilities so that these nodes may form an upper spanning tree and hence an upper communication backbone of the WSN.

Acknowledgements. This work was partially supported by Turkish Science and Research Council Career Project 104E064.

References

1. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless Sensor Networks for Habitat Monitoring. ACM WSNA (2002)
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Elsevier Comp. Ntws, **38**, 393–422 (2002)
3. Chong, J.: A Survey of Clustering Schemes for Mobile Ad Hoc Networks. IEEE Comm. Surveys and Tutorials, **7(1)**, 32–48 (2005)
4. Estrin, D., Govindan, R., Heidemann, J. Kumar, S.: Next Century Challenges: Scalable Coordination in Sensor Networks. Mob. Comp. and Networking (1999)
5. Dagdeviren, O., Erciyes, K.: A Distributed Backbone Formation Algorithm for Mobile Ad hoc Networks. ISPA06, Springer Verlag LNCS, **4330**, 219–230 (2006)
6. Younis O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad hoc Sensor Networks. IEEE Trans. on Mob. Comp., **3(4)** (2004)
7. Heinzelman, W. R., Chandrakasan, A., Balakrishnan H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. IEEE HICSS (2000)
8. Ye, F., Zhong, G., Cheng, J., Lu, S., Zhang, L.: PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. IEEE ICDCS (2003)
9. Xu, Y., Heidemann, J., Estrin, D.: Geography Informed Energy Conservation for Ad Hoc Routing. ACM/IEEE MOBICOM, 70–84 (2001)
10. Yu, Y., Govindan, R., Estrin, D.: Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. UCLA Computer Science Department TR UCLA/CSD-TR-01-0023 (2001)
11. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A Two-tier Data Dissemination Model for Large-Scale Wireless Sensor Networks. ACM MOBICOM (2002)
12. Gallagher, R. G., Humblet, P. A., Spira, M.: A Distributed Algorithm for Minimum-Weight Spanning Trees. ACM Trans. on Prog. Lang. and Sys., **5** (1983)
13. Awerbuch, B.: Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election and Related Problems. ACM STIC (1987)
14. Banerjee, S., Khuller, S.: A Clustering Scheme for Hierarchical Routing in Wireless Networks. University of Maryland Tech. Report CS-TR-4103 (2000)
15. Zou, S., Nikolaidis, I., Harms, J. J.: ENCAST: Energy-Critical Node Aware Spanning Tree for Sensor Networks. IEEE CNSR, 249–254 (2005)