Towards Potential-Based Clustering for Wireless Sensor Networks *

Leonidas Tzevelekas
University of Athens, Greece
ltzev@di.uoa.gr

Artur Ziviani
LNCC, Brazil
ziviani@lncc.br

Marcelo Dias De Amorim
LIP6/CNRS, France
Marcelo.Amorim@lip6.fr

Petia Todorova
Fraunhofer FOKUS, Germany
todorova@fokus.de

Ioannis Stavrakakis
University of Athens, Greece
ioannis@di.uoa.gr

Categories and Subject Descriptors: C.2.1 Network Architecture and Design.

General Terms: Algorithms, Design.

Keywords: Distributed Clustering Scheme, Wireless Sensor Networks.

1. CONTEXT

Wireless Sensor Networks (WSNs) are expected to operate in an autonomous fashion, thus being capable of self-assembly and continuous self-organization in an efficient, reliable, and scalable manner during their network lifetime. In general, *clustering* in wireless sensor networks provides advantages such as scalability, improved robustness, and efficient power consumption.

Although previous clustering approaches, such as [2, 4] show some interesting behavior, they overlook a fundamental parameter that has a strong impact on system performance: the cluster size. An alternative solution to form bounded-size clusters with low message overhead and low cluster overlap is through *budget-based clustering* [1]. In budget-based clustering algorithms, an initiator node (the cluster-head) is assigned an initial budget. The cluster-head starts then to distribute the budget among its neighbors, which in turn do the same until the budget is exhausted. This approach allows cluster growth to be based on local decisions rather than involving the clustering initiator at each round, thus limiting the message overhead. By controlling the allocated budget for cluster formation, one is able to control the upper bound on the cluster size. Krishnan and Starobinski [1] propose two algorithms for budget-based clustering: *rapid* and *persistent* algorithms.

In the rapid algorithm, the initiator node A is assigned a budget β_A, of which it accounts for itself and distributes $\beta_A - 1$ to its neighbors. These neighbors do the same until the budget is exhausted. Each node that receives a message sends an acknowledgement to the initiator. If a node receives a budget that it cannot propagate because there are no neighbors in its subtree, this node simply discards its extra budget. The overhead generated is then minimum, but it may result in clusters much smaller than the initial budget would allow (see Fig. 1(a)).

In order to reduce the limitations of the rapid algorithm, the persistent algorithm is proposed [1], which aims at obtaining cluster sizes as close as possible to the initial budget. When a node receives a message, it does not immediately reply to the initiator. It first computes the number of neighbors in its subtree before distributing the budget. When either the budget is met or when further growth is not possible, an acknowledgement is sent. In the case the budget cannot be completely distributed (*e.g.* because there are not enough neighbors), the node sends an acknowledgement indicating the exceeding budget. The initiator tries then to redistribute the exceeding budget. The algorithm terminates either when the budget is completely distributed or when no further growth is possible. In summary, the persistent algorithm is basically the rapid algorithm with a feedback mechanism (see Fig. 1(b)).

2. POTENTIAL-BASED CLUSTERING

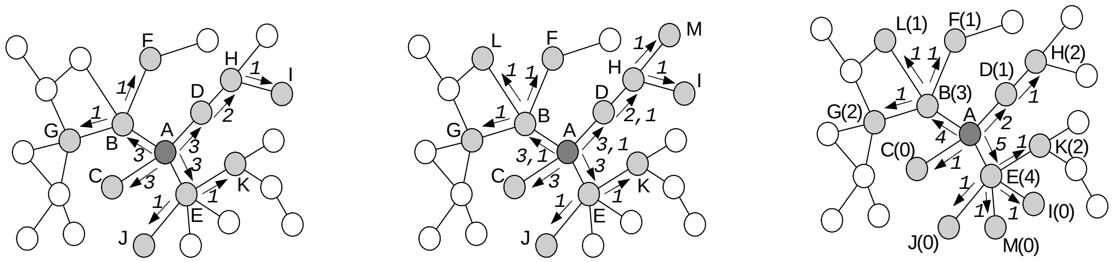
Budget-based algorithms provide an efficient solution for bounding the cluster size in wireless sensor networks. We have observed, however, that both rapid and persistent algorithms apply a blind method, *i.e.* nodes distributing budgets are *oblivious of the nature of their neighbors*. Either the nodes fully disregard their neighborhood at the risk of wasting tokens or they apply a corrective solution, which leads to an undesirable load in the network.

In this context, we propose a new methodology for on-demand determination of initiators and for obtaining a more intelligent distribution of the allocated budget. We argue that it is possible to define an algorithm that combines simplicity and low-overhead by simply learning information about neighbors. The idea is to make nodes take into account the characteristics of their neighbors before distributing budgets. We assume that nodes exchange `Hello` packets and these packets are used by the nodes to learn from each other. The algorithm we propose in this paper relies on the following assumption.

ASSUMPTION 1. *If node A has more neighbors than node B, then the subtree associated to A is likely to be larger (in numbers of nodes) than the subtree associated to B.*

*This work has been supported in part by the Greek Ministry of Education under the research Program PYTHAGORAS and the European Network of Excellence E-NEXT

CoNEXT'05, October 24–27, 2005, Toulouse, France.
ACM 1-59593-035-3/05/0010.



(a) Rapid (b) Persistent (c) Potential

Figure 1: Example comparing the three approaches. An arrow with multiple values stands for the budgets distributed in different rounds. In (c), the potential of a node is represented between parenthesis.

We call the number of neighbors of a node the *node potential*. Every node maintains a small table containing the potentials of its neighbors and uses this table to determine the amount of budget assigned to each one of them. For any node X in the network, let $\mathbf{N}(X) = \{x_1, x_2, \ldots\}$ denote the set of X's neighbors and $|\mathbf{N}(X)|$ be the size of this set. The potential of node X, denoted $\pi(X)$, is given then by

$$\pi(X) = |\mathbf{N}(X)|. \tag{1}$$

Let A be a node handling some budget β_A to be distributed among its neighbors. The amount of budget assigned to neighbor $a_i \in \mathbf{N}(A)$ is given by

$$\beta_{a_i} = \left\lfloor \frac{(\beta_A - 1)\pi(a_i)}{\sum_{j=1}^{|\mathbf{N}(A)|} \pi(a_j)} \right\rfloor. \tag{2}$$

Node A first accounts for itself and then distributes the remaining tokens based on the potentials of its neighbors following Eq. 2, *i.e.* a neighbor with higher potential receives more tokens than a lower-potential neighbor. This is in accordance with Assumption 1.

From Eq. 2, we see that some tokens may be unassigned if $(\beta(A) - 1)$ is not a multiple of $\sum_{j=1}^{|\mathbf{N}(A)|} \pi(a_j)$. In this case, the remaining tokens can be evenly distributed among the neighbors with higher potentials. Let $\beta^r = (\beta(A) - 1) - \sum_{j=1}^{|\mathbf{N}(A)|} \beta_{a_j}$. For the β^r higher-potential neighbors, node A performs $\beta_{a_i} \leftarrow \beta_{a_i} + 1$ before distributing the budget. This guarantees that no tokens are wasted. The same potential-based algorithm is applied by the nodes in the subtrees until the budget is exhausted or no further growth is possible (see Fig. 1(c)).

At last but not least, an important issue concerns the cluster management. In a dynamic WSN, nodes can change location, be removed, or added. A topological change occurs when a node disconnects and connects from/to all or part of its neighbors. One example for topology change is when nodes enter sleep mode for energy saving. Modification of the cluster structure in the presence of topology changes leads to performance degradations in the network. For that reason, a cluster management procedure is required. The clustering management scheme has to be designed to keep the cluster infrastructure stable in a dynamic environment.

3. FINAL REMARKS

The proposed scheme is currently being implemented. For the analysis of the proposed potential-based clustering scheme, we chose to proceed with the essential simulation-based analysis and evaluation. The simulation environment used is an open-source, component-based, modular and open architecture simulator called Omnet++ [3]. The simulation scenario currently under investigation is a static sensor network, where the messages exchanged feature the required communication between nodes. At this early stage of implementation, the simulator is programmed to complete the following tasks: *(i)* instantiation of the wireless sensor network; *(ii)* connectivity control of all nodes in the network; and *(iii)* execution of a clustering algorithm.

At the next stages of the simulation implementation we intend to add the distributed initiation of the clustering procedures and the addition of mobility or other factors to the environment (*e.g.*, power depletion of nodes) so as to cause the required topological changes in the network. The proposed cluster maintenance algorithms will be added and their efficiency will be checked.

We plan to investigate the performance of the above algorithms parameterized vs. the *network density per communication zone*, denoted by $\mathcal{D}$, and given by the average number of nodes per communication region of each sensor node: $\mathcal{D} = N/(\pi R^2)$.

In conclusion, the potential-based clustering algorithm is expected to outperform the previous budget-based algorithms *rapid* and *persistent*, both in terms of "compactness" of the resulting clusters and in terms of speed of network decomposition, i.e. energy efficiency of the clustering process. All this will incur at the cost of adding very low overhead to the individual nodes. Our scheme features also the cluster maintenance procedures, so overall it can deliver a comprehensive clustering framework for WSNs.

4. REFERENCES

[1] R. Krishnan and D. Starobinski. Efficient clustering algorithms for self-organizing wireless sensor networks. *Ad Hoc Networks*, 2005. In print.

[2] J.-S. Liu and C.-H. R. Lin. Energy-efficiency clustering protocol in wireless sensor networks. *Ad Hoc Networks*, 3(3):371–388, May 2005.

[3] O. D. E. S. System. http://www.omnetpp.org.

[4] O. Younis and S. Fahmy. Distributed clustering for ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. of the IEEE Infocom'2004*, Hong Kong, Mar. 2004.