A new dominating tree routing algorithm for efficient leader election in IoT networks

Ahcène Bounceur, Madani Bezoui, Massinissa Lounis, Reinhardt Euler, Ciprian Teodorov

A New Dominating Tree Routing Algorithm for Efficient Leader Election in IoT Networks

Ahcène Bounceur[1,2], Madani Bezoui[1,3], Massinissa Lounis[1], Reinhardt Euler[1,2], Ciprian Teodorov[4]
[1]Lab-STICC CNRS UMR 6285
[2]Université de Bretagne Occidentale, Brest, France
[3]Department of Mathematics
Université de Boumerdes, Boumerdes, Algeria
[4]ENSTA Bretagne, Brest, France
Ahcene.Bounceur@univ-brest.fr

***Abstract*—A leader node in Ad hoc networks and especially in WSNs and IoT networks is needed in many cases, for example to find a node with minimum energy or situated on the extreme left of the network. For this kind of applications, algorithms must be robust and fault-tolerant since it is difficult and even impossible to intervene if a node fails. Such a situation can be catastrophic in case that this node is the leader. In this paper, we present a new algorithm, which is based on a tree routing protocol. It starts from local leaders which will start the process of flooding to determine a spanning tree. During this process their value will be routed. If two spanning trees meet each other then the tree routing the best value will continue its process while the other tree will stop it. The remaining tree is the dominating one and its root will be the leader. This algorithm turns out to be low energy consuming with reduction rates that can exceed 85%. It is efficient and fault-tolerant since it works in the case where any node can fail and in the case where the network is disconnected.**

***Index Terms*—Wireless Sensor Network, IoT, Leader Election, Distributed algorithms, Dominating Tree Routing**

I. Introduction and related work

This paper comes within the context of secured sites where one needs to find the boundary nodes of wireless sensor and IoT networks. Many algorithms exist in the literature. A recent algorithm, called D-LPCN [1], can be used for this purpose. This algorithm starts from the node which is on the extreme left of the network, that we suppose to be embedded in the plane with nodes being identified by their coordinates. To find this particular node, one can use any Leader Election algorithm, which is a process of electing one particular node in a network. Usually, the leader process is required to play a particular role for coordination or control purposes. There is no solution for this problem in the case of anonymous systems [2], i.e., systems with nodes having no identifiers. In other words, there is no way to differentiate a process p_i from another process p_j. Due to this problem, we assume in this paper that each process p_i has an identifier id_i, and that any specific process has a unique identifier. Moreover, it is assumed that the identifiers can be compared with each other.

In the literature, one can define two main families of leader election methods. The leader election for ring topologies and the leader election for arbitrary topologies (e.g., ad hoc networks).

In this paper, we propose a new algorithm for arbitrary networks. This algorithm starts with a given set of local minima, each of which will start as a root the process of flooding in order to determine a spanning tree on which to route its value. If two spanning trees meet, the one routing the better value will continue the flooding process and the other one will stop it. After a given time, only one spanning tree will remain and its root will be the leader.

The remainder of the paper is organized as follows: Section II introduces the Local Minima Finding algorithm. Section III is dedicated to the proposed algorithm. The simulation results are presented in Section IV. Finally, Section V concludes the paper.

II. The Local Minima Finding (LMF) algorithm

A local minimum node, also called *Local Leader*, is the node which has no neighbor with a value smaller than its own value. But, this value is not necessarily a global minimum.

The Local Minima Finding (LMF) Algorithm uses the same principle as the previously presented *MinFind* algorithm to determine if a node is a local minimum or not, with the exception that each node will send its coordinates only once, and after reception of messages from all its neighbors, it decides if it is a local minimum or not in case it has received a smaller value than its own.

III. The proposed algorithm

The proposed algorithm is based on a tree routing protocol. It starts from local leaders which will run, as a root, the process of flooding [3] to determine a spanning tree. During this process the value of the leader (root) will be routed. If two spanning trees meet each other then the tree routing the value of the better value will continue its process while the other one will stop it. In the following, we present the main steps of the proposed algorithm, where we assume that the leader is the node having the maximum value:

1) *Step 1:* We run the LMF algorithm to determine the local minima.
2) *Step 2:* Each local minimum will start the flooding process to route the leader value (local minimum) over the tree.

3) *Step 3:* If two trees meet, the flooding process will be continued only by the nodes having the better values.
4) *Step 4:* Each local minimum will wait for a given time, assumed to be sufficient to finish the process of flooding. If after this time there is no received message anymore, the corresponding local minimum will become the leader.

IV. Simulation results

For the simulation, we have used the simulator CupCarbon [4], [5], which is a Smart City and Internet of Things Wireless Sensor Network (SCI-WSN) simulator. Its objective is to design, visualize, debug and validate distributed algorithms for monitoring, tracking, collecting environmental data, etc., and to create environmental scenarios.

To compare our algorithm with the classical *MinFind* algorithm, we have generated 9 networks in a rectangular area of $(z \times z)m^2$, where z is varied from 200 to 1000 with n randomly generated nodes. The value of n is fixed so that the density of the nodes in each network remains the same. We have fixed it to 10 $nodes/hm^2$ (hm: hectometer), i.e., 10 nodes in an area of $100m \times 100m$. Note, that we consider symmetric communications between nodes.

We have considered two cases. In the first case (case 1), each node generates a value representing its x-coordinate. In the second case (case 2), the considered value represents a random value. For each network, we have calculated the number of transmitted and received messages (exchanged messages) in order to compare their energy consumption which is directly related to this metric. We have obtained the graph of Figure 1. In both cases, we have executed the algorithm MinFind. The obtained results are shown by the black curves of Figure 1 labeled as MinFind1 (case 1) and MinFind2 (case 2) and the red curve for our proposed algorithm (case 1) and the blue curve for our proposed algorithm (case 2). As we can see, the proposed algorithm is less energy consuming than the MinFind algorithm. This is confirmed by Figure 2 which shows the reduction rate between the MinFind algorithms for both cases defined above. In the case where the leader represents the smallest x-coordinate value, we can see that the reduction rate reaches 83% for a network with 1000 nodes and it is growing for larger networks. This kind of leader is needed in the case of the D-LPCN algorithm [1] which starts from the node situated on the extreme left. In the case of other kinds of leaders (random, id, etc.) the proposed algorithm can reach a reduction rate of 30%. Altogether, we can conclude that the proposed algorithm is less energy consuming than the classical algorithm MinFind.

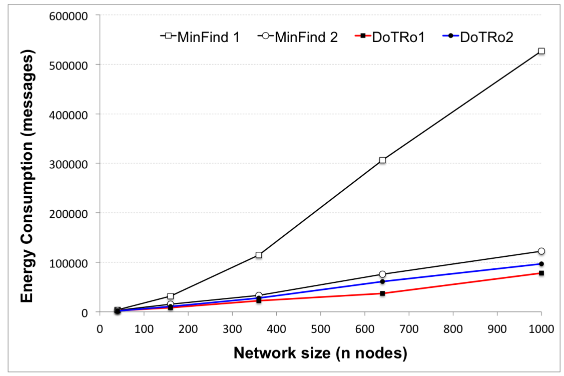

Fig. 1. Simulation results.

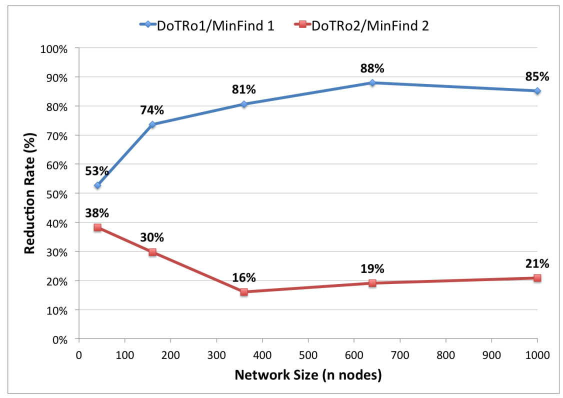

Fig. 2. Reduction rate.

V. Conclusion

We have presented a new algorithm for leader election which starts from the local minima found using the LMF algorithm. Then each local minimum starts as a root the process of flooding in order to determine the spanning tree while routing its value over it. If two spanning trees meet, the one routing the better value will continue the flooding process and the other one will stop it. After a given time, only one spanning tree will remain and its root will be the leader. The obtained results show that the proposed algorithm is less energy consuming with rates that can exceed 85% when searching the minimum x-coordinate and 30% when searching the minimum random value. Another advantage of the proposed algorithm is that it is fault tolerant since it starts even when there are failing nodes and it also finds the leader of each connected component of the network.

Acknowledgment

This project is supported by the French National Research Agency ANR PERSEPTEUR - REF: ANR-14-CE24-0017.

References

[1] Saoudi, M., Lalem, F., Bounceur, A., Euler, R., Kechadi, M. T., Laouid, A., Madani, B., and Sevaux, M., D-LPCN: A Distributed Least Polar-angle Connected Node Algorithm for Finding the Boundary of a Wireless Sensor Network, Ad Hoc Networks, Elsevier, Volume 56, 1 March 2017, Pages 56-71.

[2] M. Raynal. Distributed algorithms for message-passing systems, volume 500. Springer, 2013.

[3] Tanenbaum, Andrew S.; Wetherall, David J. (2010-03-23). Computer Networks (5th ed.). Pearson Education. p. 368-370.

[4] CupCarbon simulator, http://www.cupcarbon.com

[5] Mehdi, K., Lounis, M., Bounceur, A., and Kechadi, T. CupCarbon: A Multi-Agent and Discrete Event Wireless Sensor Network Design and Simulation Tool, In IEEE 7th International Conference on Simulation Tools and Techniques (SIMUTools'14), Lisbon, Portugal, 2014.