A New Distributed Algorithm for Finding Dominating Sets in IoT Networks under Multiple Criteria

Madani Bezoui, Ahcène Bounceur, Reinhardt Euler, Moulaï Mustapha

A New Distributed Algorithm for Finding Dominating Sets in IoT Networks under Multiple Criterias

Madani Bezoui[1], Ahcène Bounceur[2], Reinhardt Euler[2], Mustapha Moulaï[3]

[1]Department of Mathematics, University M'hamad Bougara of Boumerdes, Boumerdes, Algeria
{mbezoui}@univ-boumerdes.dz

[2]Lab-STICC UMR 6285, CNRS, Université de Bretagne Occidentale, Avenue Victor Le Gorgeu, 29238, Brest, France
{Ahcene.Bounceur, Reinhardt.Euler}@univ-brest.fr

[3]Department of operations research, Faculty of Mathematics, University of Sciences and Technology Houari Boumediene, Algiers, Algeria
{mmoulai}@usthb.dz

Keywords: *Dominating Set, Multi-criteria decision making, Wireless Sensor Networks, Internet of Things, Iterative local aggregation method.*

1 Introduction

The various tasks of everyday life have never been easier thanks to wireless sensor networks that are continually expanding in many areas, like forest fire detection [4], harvesting [8], intrusion detection [2], [10], etc. These networks are composed of a set of microsensors called nodes, which can collect and transmit data autonomously. These microsensors are often deployed randomly in a region of interest. This randomness makes it difficult to determine the topology of the network.

Energy and efficiency are the most recurrent problems in wireless sensor networks. Since energy is a resource, often non-renewable and limited, it is of great interest to preserve it and to optimize its use. The effectiveness of a wireless sensor network, and its ability to perform the task assigned to it, is part of the quality of service of this network. These two objectives are mutually contradictory and require a compromise to ensure a quality network that can serve for a long duration by minimizing the energy.

The authors of [1] propose a centralized algorithm for the construction of a connected dominating set based on multipoint relays. The authors of [7] propose a new distributed greedy algorithm for approximating the construction of a minimal connected dominating set. The authors claim that their method obtains good results in terms of construction cost compared to classical algorithms. The authors of [5] propose a system of clustering, called EBDSC, based on dominating sets that effectively extend the lifetime of the network by balancing the energy consumption between different nodes.

The main idea of this work is to detect a connected dominating set, which will be designated to perform special tasks and serve other nodes of the network. This strategy aims to reduce communication overhead and increase bandwidth efficiency, reduce overall energy consumption, and ultimately to extend the effective lifetime of the network. For this purpose, we introduce a new distributed algorithm for the search of the nodes forming connected dominating set in a wireless sensor network, using a new concept called *wait before start*, which allows each dominating node to declare itself the number of its neighbors and its residual energy. This method is based on a delayed start of each sensor with respect to an associated value. If the value refers to the number of neighbors, then the sensor having the maximum number of neighbors so far, will start the process first, so that its waiting time for an associated value corresponds to the energy consumed up to the instant t, and so on for all the nodes of the network. This procedure is hybridized with the iterative local aggregation method proposed by [9]. The results of the simulation on several randomly generated networks show that the proposed method guarantees the generation of a dominating set using two criteria: the number of neighbors and the remaining energy.

2 Problem formulation

The dominating set problem in a graph representing a wireless sensor network consists in finding the minimum number of nodes that communicate with all the other nodes of the network. For example, the set of red nodes in the Figures 1(b) or (c) represents a dominating set of minimum size because it is not possible to find a set with less than two nodes which are connected to all other nodes. However, the set of red nodes in Figure 1(a) is a dominating set but not of minimum size.

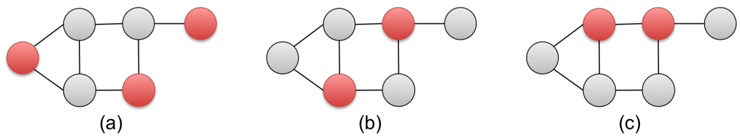

FIG. 1: Dominating sets.

2.1 Definition of criteria (attributes)

The two criteria defined in this work are:

Maximizing the total degree of the dominating set (z_1): This criterion is very important to optimize the realization of the various tasks assigned to the network. In the proposed algorithm we interpret this criterion by choosing vertices of greater degrees.

Maximizing network lifetime (z_2): The lifetime of a network is ensured by the batteries of the sensors. Optimizing this objective means to avoid the use of sensors whose battery level is too low.

Note that these two criteria are contradictory, because a user tends to use the same nodes (those covering the greatest number of other nodes). But this implies an energy consumption that is not uniform. Thus the lifetime of the network will not be optimal.

2.2 Definition of actions (alternatives)

In this work, we have to carry out the selection of a dominating set in a periodical way to ensure the two criteria defined previously. Consequently, the set actions is defined by the nodes of a network.

3 The iterative local aggregation method

This method is introduced by [9], it can be summarized as follows:

a research phase: exploitation of information to integrate the preferences of the decision-maker,

a reset phase: elaboration of a new proposal to be submitted to the decision-maker,

a phase of reaction: during this phase the decision-maker is directly involved in the decision-making process and issues an opinion on the latest proposal. His opinion is taken care of by the analyst.

The satisfaction of the decision-maker is seen as a condition for stopping this process. In the IoT networks, in the beginning, the residual energy is the same in each node. Therefore the second criterion does not have a significant influence on the selection. It is after a period (often defined statistically) noted τ that the energy factor will have its importance. We then proposed the hanging:

weight of the attribute z_1**:** $w_1 = E(d_max)$,

weight of the attribute z_2**:** $w_2 = y_1(1 - E(d_max))$,

where:

$E(d_max)$ is the residual energy of the node having the maximum degree in the network, and

$y_1 = 1$ if $t > \tau$ and equal to 0 otherwise.

4 Functions and message primitives

The distributed algorithms presented in this work use some primitive functions that we describe in Table 1.

TAB. 1: Functions of the proposed algorithms.

Function	Definition	Function	Definition
send(m,*)	sends the message m in a broadcast	rand(a,b)	generates an integer random value between a and b exclusive
read()	waits for receipt of messages. This function is blocking. If there is no received message any more, it remains blocked in this instruction	read(wt)	waits for receipt of messages. If there is no received message after wt milliseconds then the execution will continue and go to the next instruction
getTime()	returns the local time of a node	getId()	returns the node identifier
getNNeig()	returns the number of neighbors of the node	stop()	stops the execution of the program

5 The WBS algorithm

This problem is an NP-Hard problem [3], which means that no algorithm can solve it in polynomial time. In this work, we will use a classical heuristic described by Algorithm 1. To illustrate the operation of this algorithm, we first consider a network with a set of nodes V. The algorithm selects a node of maximum degrees and adds it to the set D (the current dominating set). Then, the selected node and its neighbors are eliminated from V. This procedure is repeated on the resulting graph as many times as necessary until V is empty. The set D obtained this way clearly represents a dominating set of the network.

Data: V
Result: D
$D = \emptyset$;
while *($S \neq \emptyset$)* **do**
  $n = \underset{s \in S}{\text{argmax}}(N(s))$; $e = \underset{s \in S}{\text{argmax}}(N(s))$;
  $D = D \cup \{n\}$; $V = V - N(n)$; $V = V - \{n\}$;
end

Algorithm 1: The pseudo-code of the centralized heuristic of finding the dominating set

We use a new concept called WBS (Wait-Before-Starting), in which the start time of each node depends on its value. The value chosen for this work is the number of neighbor of each node weighted with its residual energy. The weighting is carried out by an iterative local aggregation method, i.e., in the life cycle of an IoT network, the energy criterion brings little, since the majority of the batteries of the sensors are full. The process used in this work gives an increasing weight to the energy criterion, as our network works and loses energy. The pseudo-code of the WBS concept is given by Algorithm 2.

Data: x, gt
Result: D
once = false; $t = x \times gt$;
while *(true)* **do**
  r_x = read(t);
  if *(r_x==null)* **then**
    send(A, *); stop();
  end
  if *(r_x==A and once==false)* **then**
    once = true; send(A, *); stop();
  end
end

Algorithm 2: *WBS*: The pseudo-code of the WBS algorithm

6 Proposed distributed version and results

In this section, we will present the multicriteria and a distributed version of the heuristic presented in the previous section. It works as follows: each node performs the weighting explained in Section 3, and then we use the WBS concept to start the algorithm with the sensor node having the maximum weighted value. It will be admitted as a dominating node. Then, it deactivates its neighbors by sending a message T1, this step corresponds to lines 4, 5 and 6 of the Algorithm 1, then it deactivates itself. This process is repeated until $V = \emptyset$. This procedure is reproduced from the weighting periodically to find all compromise solutions. Algorithm 3 shows the pseudo-code of this process which represents the distributed version of the heuristic presented by Algorithm 1.

Data: $G = (V, E)$
Result: D

```
id = getId();   n = getNNeig();   b = rand(1, 100); once1 = false;   once2 = false;
w = (50−n)*100+b;  ts = getTime();  te = ts+w;  dominating = false;
while (true) do
    loop  wait w;  t = getTime();   read message;
    if (message==null) then
        dominating = true; send(T1);   stop();;
    else
        if ((message==T1) and (once1==false)) then
            once1 = true;   send(T2);   stop();
        end
    end
    if ((message==T2) and (once2==false)) then
        once2 = true;   n = n − 1  b = rand(1, 100);
        nw = (50−n)*100+b;   w = ((nw−w)+(te−t));
    end
end
```

Algorithm 3: The pseudo-code of the distributed heuristic for finding the dominating set

For the implementation of this algorithm, we used the CupCarbon [6] simulator. Figure 2 shows an example of a dominating set (yellow nodes) obtained by the proposed method.

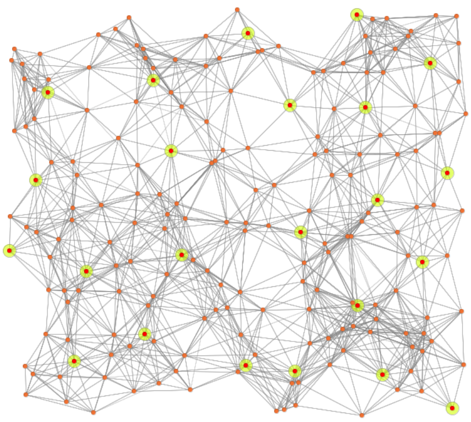

FIG. 2: An example showing 23 nodes dominating a total of 200 nodes.

7 Conclusion

In this work, we have presented a new distributed algorithm for the dominating set problem in an IoT network under several screens. This algorithm uses the Wait-Before-Starting (WBS) concept to independently determine the dominating nodes and the iterative local weighting method. In this concept , and each value is time-weighted, each node in the network will begin executing its program after a specified time associated with its value. In this work, this value represents the number of neighbors of a node and its residual energy.

Acknowledgment

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

References

[1] Cedric Adjih, Philippe Jacquet, and Laurent Viennot. *Computing connected dominating sets with multipoint relays.* PhD thesis, INRIA, 2002.

[2] Farooq Anjum, Dhanant Subhadrabandhu, Saswati Sarkar, and Rahul Shetty. On optimal placement of intrusion detection modules in sensor networks. In *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pages 690–699. IEEE, 2004.

[3] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete applied mathematics*, 23(1):11–24, 1989.

[4] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[5] Xiaoyan Kui, Jianxin Wang, Shigeng Zhang, and JLANNONG CAO. Energy balanced clustering data collection based on dominating sets in wireless sensor networks. *Adhoc & Sensor Wireless Networks*, 24, 2015.

[6] Kamal Mehdi, Massinissa Lounis, Ahcène Bounceur, and Tahar Kechadi. Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool. In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, pages 126–131. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[7] Jasaswi Prasad Mohanty, Chittaranjan Mandal, and Chris Reade. Distributed construction of minimum connected dominating sets in wireless sensor networks using two-hop information. *Computer Networks*, 123:137–152, 2017.

[8] Jayanthi Rao, Tao Wu, and Subir Biswas. Network-assisted sink navigation protocols for data harvesting in sensor networks. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 2887–2892. IEEE, 2008.

[9] Alain Schärlig. *Décider sur plusieurs critères: panorama de l'aide à la décision multicritère*, volume 1. PPUR presses polytechniques, Lausanne, 1985.

[10] Milica Pejanović Ðurišić, Zhilbert Tafa, Goran Dimić, and Veljko Milutinović. A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199. IEEE, 2012.