A Low-Overhead Localized Target Coverage Algorithm in Wireless Sensor Networks

Dimitrios Zorbas[1](✉) and Christos Douligeris[2]

[1] Inria Lille, Nord Europe, France
dimitrios.zormpas@inria.fr
[2] Department of Informatics, University of Piraeus, Piraeus, Greece
cdoulig@unipi.gr

Abstract. The scope of this paper is to present a low-overhead localized algorithm for the target coverage problem in wireless sensor networks. The algorithm divides the sensors into active and sleep mode nodes in order to conserve energy and extend the network lifetime. The set of active mode nodes provide full coverage to a set of targets (points) in the field. The decision of which sensors will remain active at any time is locally taken by the nodes by exchanging messages with each other. This kind of messages add overhead in the network, while high overhead can dramatically decrease the network lifetime especially in case of high node density environments. To tackle this problem we propose two variations of a localized algorithm with low communication complexity. Finally, the operational effectiveness of the proposed approaches is evaluated through simulation, while their superiority against other relevant proposed solutions in the literature is illustrated. The results show a great improvement in terms of communication cost while achieving an adequate network lifetime.

1 Introduction

Wireless Sensor Networks (WSNs) consist of energy constrained devices which are used in different kind of surveillance applications. One of them is the monitoring of a set of targets in the field. The targets are specific points of high interest since their coverage provides important information to the remote users. For example, in an agricultural application, a point can be a plant whose growth, soil humidity and temperature must be identified every some hours. In a military application, a point can be a piece of road where the opponent passes by and must be detected.

The presence of the targets may be known to the sensor nodes before their deployment or may be communicated and updated throughout the monitoring process. In all cases, a localization method is essential for the application, especially when the nodes are deployed randomly and their location is not predefined. GPS devices are ideal for outdoor applications, since most of the time they provide an accurate enough location (1-2 meters divergence to the actual position). Considering indoor applications other GPS-free methods are required [1].

N. Mitton et al. (Eds.): ADHOCNETS 2014, LNICST 140, pp. 245–254, 2014.
DOI: 10.1007/978-3-319-13329-4_22

The energy efficiency is important in WSNs and usually localization techniques, like GPS, consume a lot of energy. However, if the nodes are considered static, the computation of the position takes places only once, thus, the energy cost is minimal. In case of mobile nodes, the position must be updated regularly, thus, energy efficient localization solution must be used [2].

Depending on the type of the application, the surveillance of a part or of all the targets is required. This work focuses on the full target coverage problem with static nodes, where the main goal is to extend the lifetime of the involved sensor nodes, while at the same time guaranteeing 100% coverage of the targets. Since the sensor nodes are battery equipped, energy-efficient protocols should be developed. Aiming at achieving energy conservation, a certain quantity of unused nodes can remain in sleep mode (very low energy consumption mode), while the rest of the available devices used to provide coverage. Smart alternation of the nodes status between the active and the sleep mode – assuring coverage at the same time – leads to energy conservation.

In order to achieve the above described goal, the nodes must be divided into a number of sets, called cover sets. Each cover set is capable of covering all monitored targets, but only one set is active at any time. There is no restriction regarding the participation of a node in multiple sets. However, the monitoring process is divided in one or more rounds and in each round the nodes elect the current active set of sensors. The duration of the rounds is known to the sensors and they use an appropriate timer to wake up.

The majority of the existing solutions in the literature are centralized or in case of a distributed algorithm the election process exhibits a high communication overhead. Centralized solutions suffer from scalability issues, while high overhead can fast waste the nodes energy. In this paper, we overtake these problems by proposing a fully localized and low overhead algorithm for the election of the active sensor nodes. Each node is capable of deciding its status by acquiring information of its 1-hop neighbors and by keeping this information in its memory.

The remainder of the paper is structured as follows. Section 2 summarizes the related work, emphasizing at target coverage algorithms. In Section 3 the proposed problem is formulated and its solution is provided in Section 4. Section 5 evaluates the proposed algorithm and compares its performance to other works in the literature. Finally, Section 6 concludes the paper.

2 Related Work

In this section, we cite the most recent works in the area of target coverage and we classify them according to the nature of the proposed algorithms.

The target coverage problem has been extensively studied in the literature as a problem of generating the maximum number of cover sets. The problem has been proven to be NP-Complete [3,4], thus, finding the optimal number of sets is a hard process with high complexity. For that reason, many suboptimal solutions with or without performance guarantees have been proposed.

A set of mathematical optimization problems related to target coverage has been presented in the following works. In [3] linear programming is used to determine the optimal number of sets, while in [5] an algorithm based on the column generation theory is introduced. An automata based solution is described in [6], while a polynomial time approximation solution is presented in [7]. Both linear and approximation solution are used in [8] to find near optimal network lifetime. Bio-inspired and genetic approaches are presented in [9] and [10], respectively.

On the other hand, a high number of heuristic algorithms which provide a very fast solution without guaranteeing that the determined solution is the optimal one, can be found in the literature. In [11], a general methodology of how to construct a centralized coverage algorithm is described, along with some fast heuristics. Energy-efficient algorithms for different variations of the target coverage problem are also presented in [12–16].

Solving the target coverage problem using a localized or a distributed approach is an important task which has not been yet thoroughly examined in the recent literature. In [17] the authors show that when the communication range is at least two time the sensing range, the target coverage problem can be solved locally. Based on this observation they propose a localized solution. Other distributed coverage techniques are presented in [18–20], however, they are out of the scope of this paper.

3 The Target Coverage Problem

Let $T_0 = \{t_1, t_2, \ldots, t_k\}$ be the set of targets and $S_0 = \{s_1, s_2, \ldots, s_n\}$ the set of sensor nodes. Each target in T_0 is covered by at least one sensor node in S_0.

Each sensor has a sensing range equal to R_s and any target lying within the circle defined by the location of a sensor and the range R_s can be monitored with high probability by this particular sensor. On the other hand, targets outside this range cannot be accurately detected (or they cannot be detected at all) and they are considered uncovered. The covered targets are kept in P_i for each i in S_0.

Moreover, each node can communicate with other neighboring nodes which lie within a range R_c and exchange messages.

Each node's initial energy is equal to l_0 and this energy can be spent by participating in one or more generated cover sets. We assume that the energy cost during the sleep mode is negligible compared to the coverage cost in active mode. The maximum number of times each node can participate in the rounds is w ($w \in \mathbb{N}^*$) and in every round each active node consumes l_0/w amount of energy.

The coverage algorithm produces a collection $C = \{C_1, \ldots, C_m\}$ of m cover sets. Each cover set C_p is a subset of the available sensors ($C_p \subseteq S_0$) and covers all targets found in T_0.

The main objective of a coverage algorithm is to extend the lifetime of the network by maximizing $|C|$, where $|C|$ is the cardinality of the generated collection C of cover sets. The theoretical maximum number of sets is computed by the product $|min\_t|\frac{l_0}{w}$, where $|min\_t|$ is the cardinality of the target which is covered by the minimum number of sensors in the network.

4 The LOLOCA Solution

In this section, we introduce "LOLOCA" (Low Overhead LOcalized Coverage Algorithm), an algorithm which bases its functionality on three characteristics: (a) the control of the messages and how often the nodes communicate with their neighbors, (b) the coverage status of each node's neighbors, and (c) the node's own coverage status. We present the general characteristics of LOLOCA and we distinguish two ways to handle the the poorly covered targets in the network (targets that are not covered by many nodes).

In LOLOCA, the monitoring process is divided in rounds and each round consists of the initialization and the coverage phase. The initialization phase takes place at the beginning of each round and the nodes elect a set of nodes who will be active and provide coverage until the next election. The initialization phase is short and its duration is considered negligible compared to the coverage period.

In the following lines we describe how the nodes control the communication with their neighbors, how they elect the active nodes, and they handle the poorly covered targets.

4.1 Neighbor Discovery and Coverage Status

LOLOCA considers four types of nodes regarding their election status. Nodes that have been elected as active and have sent their status to their neighbors, have status equal to 2. Nodes that have been elected as active but they have not sent yet their decision to the other nodes, have status equal to 1. Nodes that are still during their decision have status equal to 0, while nodes that will remain in sleep mode during the round have status equal to -1.

It is obvious that at the beginning of the process, all the nodes have status equal to 0. This status can change to 1 if the node is elected as active or to -1 if a node decides to go sleeping. The status can change from 1 to 2 once an active node has communicated with its neighbors and declared his status.

The nodes are allowed to communicate with other nodes and exchange their status and their P sets in two cases:

- at the beginning of the first round while their election status is 0,
- their status is 1.

It practically means that, each node can communicate at most two times during the construction of the first cover set and only once for each of the next rounds.

Each node i in S_0 keeps the received messages in set N_i. This set is kept in node's memory. Each received message contains the list of the targets the neighboring node j covers (i.e., P_j), its current lifetime and its current status. Since the node's memory may be limited, N_i is updated every time the node receives new neighboring messages and only the double entries are deleted keeping only the most recent ones (the ones with the higher election status).

Based on the sets in N, each node keeps control of its own coverage status and the coverage status of its neighbors. Each time a node receives a message from a neighbor who has been elected as active, it updates each coverage status and the coverage status of its neighbors in N. The coverage status is used later in the computation of the contribution of each node and the status decision.

The communication complexity of LOLOCA is $O(n + nm)$, where n is the number of sensors and m the number of rounds (generated sets).

4.2 Election Process

The election process is a local decision process where each node decides if it will be active or in sleep mode during the current round. Each node compares its own contribution against the other nodes in the neighborhood, thus, the information kept in N is crucial during this process.

The node with the highest contribution in the neighborhood will be active in this round, will set its election status to 1, and will declare its decision to its neighbors. On the contrary, the rest of the neighboring nodes (nodes with lower contribution) will wait until they receive a message, will update their N set and will recompute their contribution. If all the targets in P have been covered by other elected nodes, then the node will remain in sleep mode. The initialization phase terminates when all the nodes have decided to remain active or in sleep mode.

The decision of staying active or not can be taken in two different ways. Considering the first way, each node computes its own contribution as well as the contribution of its neighbors with which it has common covered targets. Based on this approach, each node knows who will be active in this round and it just waits to receive a message from the active node and recompute its contribution. An alternative way to decide a node's status is to compute a waiting time before it declares itself active or inactive node and sends a message to its neighbors. This waiting time is higher for low contribution values and vice versa. If the node does not receive any message from neighboring nodes during this waiting period, it declares itself as active, otherwise it remains inactive and recomputes the contribution formula. The first method requires some more computations from the nodes as well as predefined communication slots during the initialization phase. The second way may cause synchronization issues due to network delays (buffer delays etc.).

4.3 Contribution and Poorly Covered Targets

The contribution of a node is measured using a cost function. The cost function handles the coverage status of the nodes and their association with the poorly covered targets. Since this type of targets sets an upper limit on the maximum possible generated number of cover sets, the cost function should be aware of avoiding double-covering such targets during a single round.

LOLOCA distinguishes two ways to deal with poorly covered targets. We name the two instances "Critical" and "Badness" which are based on the works presented in [21] and [11] respectively.

In LOLOCA-Critical, the poorly covered targets are handled by computing the most critical ones (the ones covered by the minimum number of nodes [15]). This process is repeated every time a target in P_i is covered by another node with higher status in N_i. Note that the critical targets may be more than one and all of them are taken into account. Each node i computes the cost function described by Formula (1).

$$CF_i = |P_i| - |PN_i| + \frac{l_i}{l_0} + \frac{i}{\texttt{max\_int}} \tag{1}$$

where $|PN_i|$ denotes the number of targets in P_i that are already covered (by other nodes in N). The term $\frac{i}{\texttt{max\_int}}$ is used to avoid having two nodes with the same contribution. Node i uses the same formula to compute the contribution of its neighbors. Only neighbors with at least one common target with i are used. The corresponding P_j, PN_j, and l_j for each neighbor j are taken from N_i.

Additionally, LOLOCA-Critical penalizes the nodes which cover critical targets by reducing their contribution by x points for each critical target they cover ($x >> CF$). Node i's contribution is penalized only for the already covered targets in P_i (by nodes in N_i). This action promotes nodes with no association with critical targets.

On the other hand, LOLOCA-Badness evaluates the nodes' association with poorly covered targets by assigning a weight to each node called "badness" and is represented by b_i [11]:

$$b_i = \sum_{p=1}^{|P_i|} (\mu_i - |t_p| + 1)^3, \tag{2}$$

where μ_i is the maximum target cardinality found in the neighborhood of i and $|t_p|$ denotes the cardinality of target t_p.

Badness is higher for nodes which cover poorly covered targets and vice versa. Involving its value into the cost function the node's contribution can be affected:

$$CF_i = \alpha \frac{|P_i| - |PN_i|}{|PN_i| + 1} + \beta(1 - \frac{b_i}{max\_b_i}) + \gamma \frac{l_i}{l_0} + \frac{i}{\texttt{max\_int}}, \tag{3}$$

where α, β are coefficients whose value is predefined [11] and max_b_i is the maximum badness in the neighborhood of i.

The weakness of LOLOCA-Badness is that each node must broadcast its badness value to its 1-hop neighbors once it has computed it. To tackle this weakness, each node can compute the badness for each other node in the neighborhood by using its N set. However, this can cause network instability and deadlocks since the exact cardinality of targets outside the neighborhood is not known to the nodes.

Both LOLOCA-Critical and LOLOCA-Badness elect the node with the highest contribution in the neighborhood. The elected node broadcasts its election status to its neighbors and they consequently update their N set. Nodes whose all the targets in P are already covered, change to sleep mode, while the rest of the nodes continue with a new evaluation. Apparently, nodes with no remaining energy do not take part anymore in the next elections. We assume that the last election (round) takes place when at least one target is not covered anymore by any sensor.

5 Evaluation and Discussion of the Results

In this section, we evaluate the proposed solutions and we compare their performance to a centralized approach called "Dynamic-CCF" [11] and to a localized one called "DOCA" [17]. Specifically, we measure the number of generated sets (network lifetime), the number of sent messages (overhead), and the average execution time per node and per generated set (computation cost).

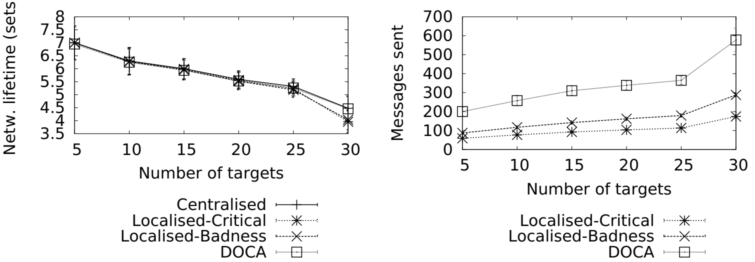
We evaluate the algorithms in two scenarios. In the first scenario, we keep constant the number of nodes and we vary the number of targets, while in the second scenario, we keep constant the number of targets and we vary the number of sensors. In these two cases, we assess the target or sensor density on the algorithms' output. Each instance of the simulation has been executed 50 times and the average results are presented as well as the 95% confidence intervals.

The terrain area is fixed and equal to 100x100m, the sensing range is 10m, and the communication range is 30m. All the targets and sensors are randomly deployed in the area using the uniform distribution. α, β and γ coefficients of LOLOCA-Badness are set equal to 0.4, 0.1, and 0.5, respectively. The maximum number of node participations is set to 2. All simulations were carried out on a Intel Xeon 2.67Ghz host, running the Debian GNU/Linux operating system, and no parallel processing of the same instance was allowed.

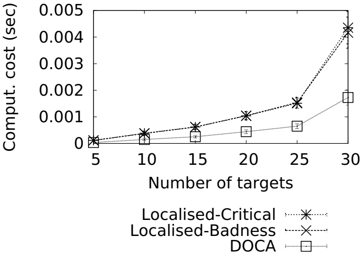
5.1 Numerical Results

In the first scenario, the network consists of 200 static nodes and variable number of targets. The results are illustrated in Figure 1. The first figure shows that the number of generated sets decreases as the number of targets increases, since more sensors remain active to cover all the targets. The produced network lifetime is almost equivalent for all the approaches. DOCA performs slightly better when many targets are deployed, however, it sends more than three and two times more messages than LOLOCA-Critical and LOLOCA-Badness, respectively (see Figure 1(b)). For the rest of the cases our localized solutions exhibit the same network lifetime with the centralized approach which achieved the theoretical maximum lifetime for the 99% of the instances. The overhead is also kept in very low levels. Our proposed approaches have higher computation cost, however this cost can be easily carried out by a modern node CPU unit [22].

In the second scenario, we assess the algorithms' behavior on node density by varying the number of sensors between 100 and 300 with an increment of 50 considering a fixed number of targets. The results of this simulation are presented in Figure 2. As discussed in the previous scenario, the algorithms achieve an

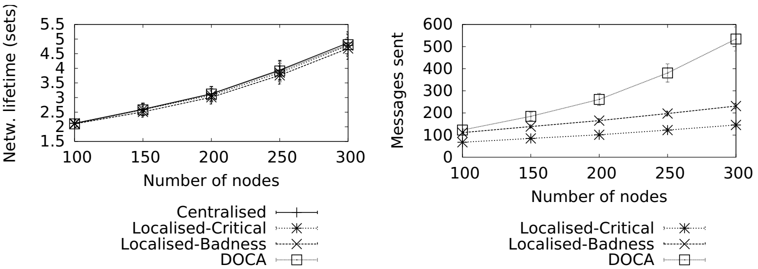


(a) Network lifetime (b) Sent messages

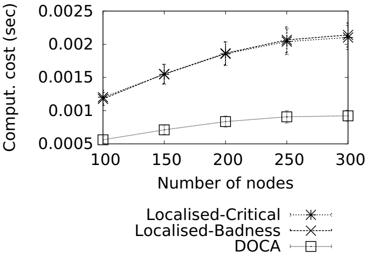


(c) Computation cost

Fig. 1. Different measurements for a scenario with 200 sensors and variable number of targets



(a) Network lifetime (b) Sent messages



(c) Computation cost

Fig. 2. Different measurements for a scenario with variable number of sensors and 20 targets

almost adequate performance in terms of network lifetime which is very close to the optimal solution. DOCA exhibits higher overhead, which is almost five times more than the competitors' one when the density is high. Finally, the computation cost of LOLOCA-Critical and LOLOCA-Badness is higher, but it is kept within reasonable limits.

6 Conclusion and Future Work

In this paper, we proposed "LOLOCA", a localized algorithm for the target coverage problem in wireless sensor networks. LOLOCA exhibits low communication cost by allowing the nodes to keep some information in their memory. We presented two instances of the algorithm using different techniques to deal with the poorly covered targets. The simulation and the comparison results showed that both instances outperform the distributed algorithm of [17] in terms of overhead, while they achieve similar network lifetime performance. However, the computation cost is higher. Our future work includes the use a real experimentation platform in order to evaluate other network characteristics like delay, synchronization, and convergence.

Acknowledgments. This work is partially supported by a grant from CPER Nord-Pas-de-Calais FEDER CIA.

References

1. Deak, G., Curran, K., Condell, J.: A survey of active and passive indoor localisation systems. Computer Communications **35**(16), 1939–1954 (2012)
2. Liu, Y., Yang, Z., Wang, X., Jian, L.: Location, localization, and localizability. Journal of Computer Science and Technology **25**(2), 274–297 (2010)
3. Cardei, M., Thai, M., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: Proc. of INFOCOM 2005, vol. 3, pp. 1976–1984. IEEE (March 2005).
4. Luo, W., Wang, J., Guo, J., Chen, J.: Parameterized complexity of max-lifetime target coverage in wireless sensor networks. Theoretical Computer Science **518**, 32–41 (2014)
5. Gu, Y., Zhao, B.H., Ji, Y.S., Li, J.: Theoretical treatment of target coverage in wireless sensor networks. Journal of Computer Science and Technology **26**(1), 117–129 (2011)
6. Mostafaei, H., Meybodi, M.: Maximizing lifetime of target coverage in wireless sensor networks using learning automata. Wireless Personal Communications **71**(2), 1461–1477 (2013)
7. Ding, L., Wu, W., Willson, J., Wu, L., Lu, Z., Lee, W.: Constant-approximation for target coverage problem in wireless sensor networks. In: INFOCOM, 2012 Proceedings IEEE, pp. 1584–1592 (March 2012).
8. Cheng, M., Gong, X.: Maximum lifetime coverage preserving scheduling algorithms in sensor networks. Journal of Global Optimization **51**(3), 447–462 (2011)
9. Ding, Y.S., Lu, X.J., Hao, K.R., Li, L.F., Hu, Y.F.: Target coverage optimisation of wireless sensor networks using a multi-objective immune co-evolutionary algorithm. Intern. J. Syst. Sci. **42**(9), 1531–1541 (2011)

10. Gil, J.M., Han, Y.H.: A target coverage scheduling scheme based on genetic algorithms in directional sensor networks. Sensors **11**(2), 1888–1906 (2011)
11. Zorbas, D., Glynos, D., Kotzanikolaou, P., Douligeris, C.: Solving coverage problems in wireless sensor networks using cover sets. Ad Hoc Networks **8**, 400–415 (2010)
12. Hongwu, Z., Hongyuan, W., Hongcai, F., Bing, L., Bingxiang, G.: A heuristic greedy optimum algorithm for target coverage in wireless sensor networks. In: Pacific-Asia Conference on Circuits, Communications and Systems, pp. 39–42 (2009).
13. He, J., Xiong, N., Xiao, Y., Pan, Y.: A reliable energy efficient algorithm for target coverage in wireless sensor networks. In: 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 180–188 (June 2010).
14. Zhao, Q., Gurusamy, M.: Lifetime maximization for connected target coverage in wireless sensor networks. IEEE/ACM Trans. Netw. **16**(6), 1378–1391 (2008)
15. Zorbas, D., Douligeris, C.: Connected coverage in wsns based on critical targets. Computer Networks **55**(6), 1412–1425 (2011)
16. Zhao, Q., Gurusamy, M.: Connected k-target coverage problem in wireless sensor networks with different observation scenarios. Comput. Netw. **52**(11), 2205–2220 (2008)
17. Zhang, H., Wang, H., Feng, H.: A distributed optimum algorithm for target coverage in wireless sensor networks. In: Asia-Pacific Conference on Information Processing, APCIP 2009, vol. 2, pp. 144–147(July 2009).
18. Guo, P., Jiang, T., Zhang, Q., Zhang, K.: Sleep scheduling for critical event monitoring in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems **23**(2), 345–352 (2012)
19. Bulut, E., Korpeoglu, I.: Sleep scheduling with expected common coverage in wireless sensor networks. Wireless Networks **17**(1), 19–40 (2011)
20. Nan, G., Shi, G., Mao, Z., Li, M.: Cdsws: coverage-guaranteed distributed sleep/wake scheduling for wireless sensor networks. EURASIP Journal on Wireless Communications and Networking 2012(1) (2012).
21. Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor networks. In: Proc. of International Conference on Communications (ICC 2001), pp. 472–476. IEEE (June 2001).
22. ST: Arm cortex-m3. http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1031/LN1565/PF164485