

Delaunay-triangulation based complete coverage in wireless sensor networks

Chinh T. Vu

Computer Science Department
Georgia State University
Atlanta, Georgia 30303
Email: chinhvtr@cs.gsu.edu

Yingshu Li

Computer Science Department
Georgia State University
Atlanta, Georgia 30303
Email: yli@cs.gsu.edu

Abstract—One of the major issues in a wireless sensor network (WSN) is how to cover an interested area. In this paper, we consider the area coverage problem for variable sensing radii WSN. With variable sensing range, the difficulties to cover a continuous space in the area coverage problem becomes exceptionally harder than covering discrete points in the target (or point) coverage problem. Very few papers have paid effort for the former problem. Wang and Medidi have solved the area coverage problem based on Delaunay triangulation structure [2]. However, due to the boundary effect their proposed algorithms cannot always provide complete surveillance for the whole network. In this work, we improve the work in [2] so that the monitored area can be completely covered. A theorem confirms that our improved algorithm provides complete coverage for all the cases. In addition, the simulation further shows that our energy-efficient algorithm has an obvious improvement on coverage status with very small compensation of network lifetime.

I. INTRODUCTION

Sensor network is a special kind of ad-hoc networks which comprises of large number of sensing, communication and computation capable devices. One of the major and most important tasks of those tiny devices is to watch over the environment (e.g., a forest) or set of subjects (e.g., collection of precious renaissance paintings) and collect environment parameters and maybe, further monitor the environment. To achieve this, the network must have surveillance over all the environment space or all the subjects. This is the objective of coverage problem. Since the introduction of coverage problem in WSN [1], researchers have tried to relax requirements to achieve better and more efficient results (network lifetime, fault-tolerance).

The other major issue in WSN is to save energy consumption and extend the network lifetime. Many work aims to balance the energy consumption among sensors as a method to extend the network lifetime.

In this work, we improve the work in [2] in order to provide complete coverage for the network while the energy consumption among sensors are balanced. We also point out and correct a mistake in this work. We assume the sensing region model of a sensor as a circle centered at the sensor's location and sensor's sensing range being the radius. We

further assume that sensor could smoothly vary its sensing range to a maximum cutoff range, denoted by $MaxR_s$. We also assume that a sensor and its neighbors have different IDs, i.e., within 1-hop vicinity, a node has unique ID.

II. RELATED WORK

The coverage problems for sensor networks can be categorized into three broad types [3] - area coverage (where every point inside an area is to be monitored), target coverage (where the main objective is to cover a set of discreet targets), and breach coverage (the goal here is to minimize the numbers of uncovered targets or the ratio of uncovered portion to the whole area). Since Wu and Yang in [7] first introduce the coverage problem with the sensor network where sensors may vary their sensing ranges, some work dedicate effort for this problem. Most of them [8], [4], [5], [6] try to solve the target coverage problem since it is easy to verify the coverage status of discreet target. Very few work [2] solve the area coverage problem. Some work [4], [5], [6], [7] assumes sensors are only capable of adjusting their sensing radii to the discreet set of some pre-defined options. Some other work [8], [2] deals with network where sensors can smoothly adjust their sensing radii to any range under some upper threshold.

The work in [7] deals with sensors having two or three levels of sensing ranges to choose. The work in [4] extends to allow sensors to vary sensing radii in one of P levels. With this relaxed assumption, the authors propose three algorithms - one ILP-based centralized, one greedy centralized and one greedy distributed - with the objective of maximizing the number of set covers, each is able to cover the whole set of targets. The work in [6] makes a step further by providing connectivity for each of those set covers.

The work in [8] proposes a centralized algorithm to cover set of targets which schedules sensors based on Garg-Könemann method. The algorithm has approximation of $(1+\epsilon)(1+\log m)$ for any $\epsilon > 0$ where m is the number of targets.

Except the work in [7], most of the work deals with target coverage problem and does not take energy into consideration. In [2], Wang and Medidi propose two distributed algorithms based on Delaunay-triangulation structure. However, even the authors have proved that the proposed algorithms could provide complete coverage, the proof still has flaw because it

does not take into account the so-called boundary effect. We will go into detail about this flaw in section III. In this work, we improve one of their algorithms such that the our new algorithm can completely cover the whole area.

The rest of the paper is organized as follows: in Section III, we explain the original algorithm introduced in [2], we then point out the flaw in their algorithm and we fix it in our proposed fixed algorithm. In the next section, we dedicate to simulation illustrating the correctness and efficiency of our algorithm. We first show the energy model and the simulation configuration, and in the next subsection we show the simulation results and explain the behind-scene meaning of those results.

III. THE IDT ALGORITHM

A. Original algorithm proposed in [2]

In this section, we will explain some key aspects of energy balancing heuristic mentioned in [2]. We name this heuristic as Original Delaunay-Triangulation algorithm (ODT). We name our heuristic as Improved Delaunay-Triangulation algorithm (IDT) which we will explain in more detail in subsection III-C.

Firstly, the authors in [2] introduce a completely distributed algorithm which requires only one hop neighbor information to construct an approximate Delaunay-triangulation. It would be necessary to emphasize that the simple algorithm proposed in [2] only constructs an approximate Delaunay-triangulation, not exact Delaunay-triangulation. Thus all the Voronoi cells or Delaunay triangles built by that algorithm are approximate. The approximate algorithm works as follows. Each sensor constructs its Voronoi constellation by drawing bisector of the line connecting it with a neighbor. Depending on the relative location of itself, the bisector line and the constellation constructed so far, the constellation will be modified accordingly. The algorithm to construct Voronoi diagram is very straight forward and intuitive. After having Voronoi diagram, the Delaunay triangulation can easily be derived by connecting pair of sensors whose Voronoi cells share a same border.

The next step, each sensor steps through all Delaunay triangles which have it as a vertex. For each triangle, the sensor calculates a weighted centroid based on the location of three vertices (which include that sensor) and their residual energy. Having the coordinate of this weighted centroid, the sensor calculates the distance from itself to the weighed centroid. The sensor radius will be the larger of its currently assigned sensing radius and that distance.

This way, all the Delaunay triangles are guaranteed to be covered by its three sensors locating at its three vertices. Since the whole area is partitioned into a set of Delaunay triangles, the whole area is covered too. However, there is still an issue which we will discuss in detail next.

B. The boundary effect issue

For coverage problem, researchers sometimes have to avoid so-called boundary effect (e.g., [10]) which is caused by the fact that the coverage status in the bordering region is different from the center region. Specially, for algorithms which are

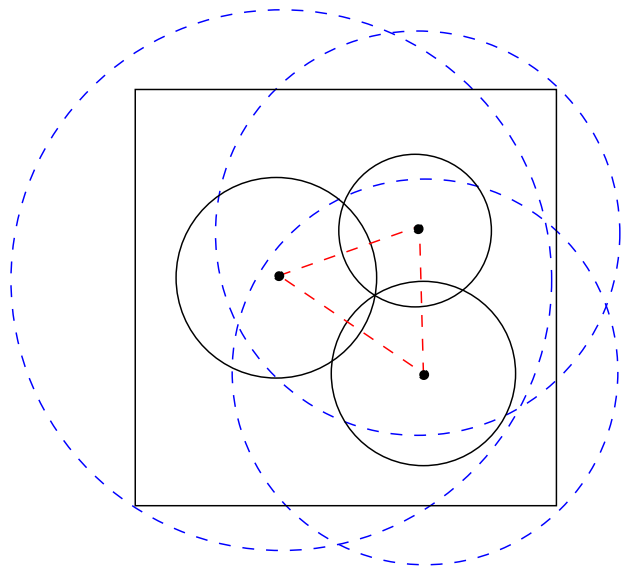


Fig. 1. Boundary effect example

based on the geographical locations of sensors, the border effect becomes even more severe since the sensors closed to the area border are not covered by neighbors in all directions.

As discussed in section III-A, we have explained the intuition behind ODT. A question arises is “what happens for sensors locating at boundary region whose Delaunay triangles are not completely surround them, i.e., there exists an angle of the sensor that does not belong to any Delaunay triangles?”

To answer the above question, let us take an example shown in Fig. 1. The dashed-lined circles represent sensing region of three sensors with their maximum sensing ranges. It can easily be seen that these three sensors completely cover the whole monitored area (monitored area is represented by the solid-bordered square). The dashed-bordered triangle is a Delaunay triangle created by three sensors. The solid-lined circles are sensing regions of them being assigned by ODT. Apparently, these sensors cannot cover the whole area with the new assigned sensing radii. The reason is that the Delaunay triangle forces those sensors to reduce their radii, therefore, they cannot reach the border. As the result, the region around the border is un-covered (or breached). Thus, with sensing range assignment scheme made by ODT algorithm, it is still possible that there exists a uncovered region. Next, we explain our improved IDT algorithm which rules out this possibility.

C. IDT algorithm

Before showing our algorithm, it is necessary to mention the k-NC rule proposed in [11] which we will employ in our algorithm. Considering the perimeter circle of sensing region of a sensor, it can be partitioned into some portions, and each portion is covered by the same set of neighbors. For each portion, the number of neighbors which cover that portion is called the perimeter coverage level of that portion. The perimeter coverage level of a node is defined in [11] as the minimum perimeter coverage level of all its perimeter

portions. This parameter usually is denoted by k . Please refer to Definition 5, in [11] for the formal definition of k -perimeter-coverage. In this work, we only consider the case where $k = 1$.

Based on that definition, the authors in [11] propose a rule which allows a sensor to verify if the network is completely covered or not based on only 1-hop neighbor information. The rule claims that if all the sensors in the network are 1-perimeter-covered, then the network is also 1-covered.

For the sake of ease to explain our IDT algorithm, we introduce some definitions:

Definition 1: (Uncovered portion) A portion of a sensor's sensing perimeter which is not covered by any of its neighbors is called an *uncovered portion*.

Definition 2: (Uncovered nodes) With current sensing radius assignment, a node is called *uncovered node* if it is not completely perimeter-covered by its neighbors.

Definition 3: (Useful neighbors) A neighbor is *useful* to a node if when being assigned with the maximum sensing ranges, it can cover the uncovered portion of that node.

The algorithm pseudo-code is shown below:

Algorithm 1 : IDT algorithm run at sensor s

- 1: Collect 1-hop neighbor information
 - 2: Build the local Delaunay triangulation based on some existing distributed algorithm like [9]
 - 3: **foreach** Triangle Δ with 3 sensors s, t, u as its vertices **do**
 - 4: \triangleright Determine weighted centroid of Δ .
 - 5: $m_x = \frac{E(s) \times s_x + E(t) \times t_x + E(u) \times u_x}{E(s) + E(t) + E(u)}$
 - 6: $m_y = \frac{E(s) \times s_y + E(t) \times t_y + E(u) \times u_y}{E(s) + E(t) + E(u)}$
 - 7: $s.R = \max(s.R, \|sm\|)$ \triangleright assign sensing range
 - 8: **end for**
 - 9: \triangleright sensors at border increase sensing ranges .
 - 10: **if** s has the highest priority among all the neighbors && NOT completely perimeter- covered **then**
 - 11: Adjust its sensing range to $MaxR_s$.
 - 12: Ask all useful neighbors to adjust their sensing range to $MaxR_s$.
 - 13: **end if**
-

After constructing Delaunay triangulation as explained in Subsection III-C, each sensor may belong to several Delaunay triangles. Consider a sensor s , it will step through all the triangles it belongs to. For each triangle, it will calculate a virtual point named "weighted centroid". The coordinates (x and y) of this point are calculated by line 5 and 6 in the algorithm 1 where $E(s), s_x, s_y$ are sensor s 's current residual energy, x coordinate, and y coordinate, respectively. t and u are other two vertices of the triangle. Given the weighted centroid's coordinates, s changes its sensing range to the distance from itself to that weighted centroid. Since there are several triangles having s as a vertex, each will have its own weighted centroid. The final sensing range of s will be the maximum distance to the furthest weighted centroid. This step to assign sensor sensing range corresponds with the heuristic for energy balancing in [2]. The work in [2], however, has

ignored the boundary effect. The ODT algorithm may leave some region uncovered. We take into account the boundary effect by letting a sensor which is not completely covered and all its neighbors who could help to cover the uncovered portion at the boundary to increase sensing ranges to the maximum value ($MaxR_s$). This step works as follows. After being assigned sensing ranges in previous step, all the sensors check if they are 1-perimeter-covered or not by using $k - NC$ rule proposed in [11]. For the ones who already are, they are just waiting for the request from their neighbors if there are any. To avoid unnecessary increase of sensors' sensing ranges at the same time, we employ the notation priority. The priority could be decided by the combination of many other parameters. In this work we define priority of a node as $\langle E_s, ID \rangle$ where E_s is sensor's residual energy and ID is sensor's ID. For a sensor that has not been 1-perimeter-covered, it firstly waits for decision (satisfied with sensing range assigned by ODT or has already increase sensing range to $MaxR_s$) from its neighbors who have higher priorities. When all those higher-priority neighbors had made their decision, the node use $k - NC$ to verify its perimeter coverage one more time. If it is still not 1-perimeter-covered, it increases its sensing range to $MaxR_s$. It then re-evaluates its perimeter coverage status (using $k - NC$ rule) to see if it is 1-perimeter-covered. If it is still not, it will ask all the useful neighbors to also increase sensing ranges to $MaxR_s$.

Next, we will prove that our improvement indeed guarantees complete coverage for the network.

Theorem 1: If the monitored area can be completely covered by the network when all the sensors are assigned maximum sensing range, then IDT ensures that the whole monitored area is completely covered.

Proof: Prove by contradiction.

Assume that after all the sensors are assigned sensing range by IDT algorithm, there still exists an uncovered region G . Theorem 4.1 in [2] has proven that the part of the area which is covered by Delaunay triangles will be covered by the sensing ranges assignment scheme of ODT, thus, G is a region that is not covered by any Delaunay triangle and G share at least one border with the monitored area.

Clearly, G shares the border with some sensors' sensing region, denoted by s_1, s_2, \dots, s_l where each s_i is a neighbor of some others and each has sensing range of $MaxR_s$ (because of line 11, 12 of algorithm 1).

Consider a point p inside G , meaning $\|s_i p\| > MaxR_s$ for all $1 \leq i \leq l$. According to our assumption that the whole area could be covered if all the sensors are assigned with the maximum sensing range, there will exist a sensor t which can cover p , meaning $\|tp\| \leq MaxR_s$, t is assigned a sensing radius smaller than $MaxR_s$, and t is a sensing neighbor of at least one sensor of set s_1, s_2, \dots, s_l . Clearly, $t \notin G$, otherwise, part of G will be covered by a Delaunay triangle. That means if being assigned sensing range of $MaxR_s$, t will cover part of uncovered perimeter portion of at least one of s_1, s_2, \dots, s_l . Or in other words, t is a useful neighbor of at least one of s_1, s_2, \dots, s_l . According to line 12 of algorithm 1, t will

increase its sensing range to $MaxR_s$, thus p is covered by t which leads to a contradiction.

Thus, the correctness of the IDT algorithm is proven. ■

IV. SIMULATION

A. Network configuration, energy consumption model

For energy consumption, we employ the quadratic energy model as in [12]. That is, when a sensor is in active mode, the amount of energy consumption for 1 unit of time is proportional to square of the sensor's sensing range. The formal formula could be as follows:

$$energy\ consumption = \frac{1}{\kappa} \times R_s^2$$

where :

- κ is a energy parameter which depends on the characteristic of a sensor.
- R_s is the sensor's sensing range.

The simulation configuration is listed in the following table:

TABLE I
SIMULATION CONFIGURATION

Parameter	Value
Network size	$800^m \times 400^m$
Number of sensors	100 ~ 500
Energy range	100 ~ 120 mJoules
Maximum sensing range	100^m
κ	8000

B. Simulation Result

We fairly run the simulation for two algorithms under the same network (i.e., same number of sensors, sensors' locations and energy assignment and all other simulation parameters are also the same). For each size of set of sensors, we execute the simulation 50 times and what is shown in below plots are the average result from those 50 simulation runs.

Fig 2 presents the comparison on average area coverage level which is measured by the average of coverage level of all the points inside the monitored area. For each point, the coverage level could be 0 (meaning that point is not covered by any sensor) or 1 (meaning that point is covered by, maybe more than 1 sensor). The plot shows the percentage of covered area out of the whole monitored area. As it can be observed, IDT completely covers the whole area for all time while ODT cannot. Moreover, the result of ODT is not stable, e.g., the network with 100 sensors provide better coverage than the one with 130 nodes. This shows that IDT is more stable than ODT.

Fig 3 measures the success rate of the two algorithms. If the result of an algorithm can completely cover the monitored area, it is a successful execution. In other words, this plot shows the percentage of successful times of ODT and IDT. As can be seen, our IDT provides complete coverage for all cases while ODT does not. Particularly, when the number of

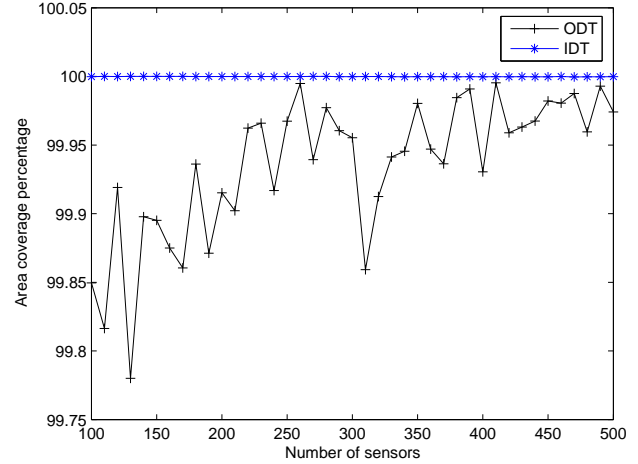


Fig. 2. Average area coverage level

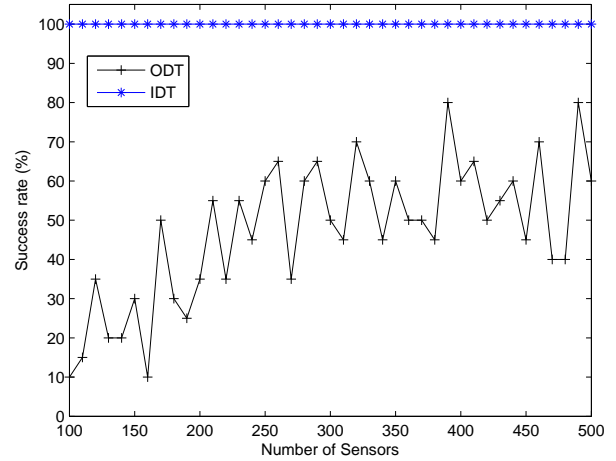


Fig. 3. Success rate (%)

nodes is 100, the chance for ODT to be successful is only 10%. This effect can easily be understood considering the boundary effect explained in Subsection III-B.

Next, we compare the network lifetime affected by the two algorithms. We first formally define the term network lifetime that is utilized in this work.

Definition 4: (Network Lifetime) The network lifetime is the duration in which the network could provide the same coverage status is maintained from the very initial beginning.

Even the above network lifetime definition is not fair for our IDT since usually ODT cannot provide complete coverage while IDT can, as shown in Fig. 4 the IDT's network lifetime is almost the same as that of ODT. Sometimes, the IDT's network lifetime is even longer. The reason is the boundary effect. That is, the upper bound of the network lifetime is limited by the lifetime of sensors close to the boundary regions since the coverage level of this area is significantly smaller than that in the central regions. Thus, if a critical sensor, the one covering

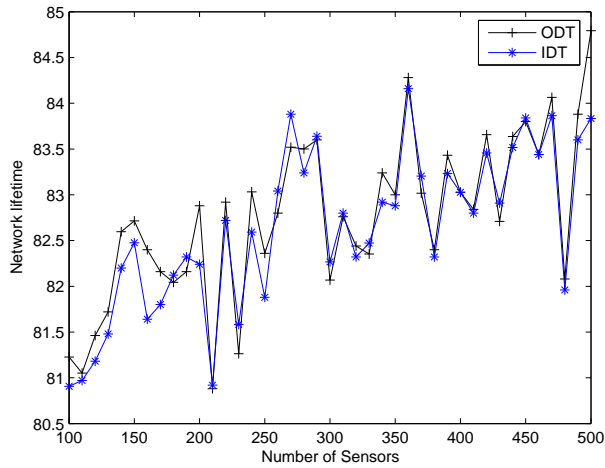


Fig. 4. Network lifetime

the portion which is not covered by any other sensors, dies, the network dies as consequence.

It would be seen that if the sensors density at the boundary area is the same as that in the center area, the performance of IDT and ODT would be the same. However, it would be hard to achieve that since sensors are randomly deployed. One way to reduce the difference between the sensors density in the boundary area and that in the center area is to deploy more sensors “across” the monitored area, i.e., near to the border and may be outside the border, the performance of ODT must improve. Apparently, the performance of IDT also enhances. We believe, even in that case, IDT will still outperform ODT at least in term of network lifetime. The reason is because of the fact that the sensor density in the center area is still higher than that in the boundary area, thus after those sensors who “close” to the borders die, the network created by ODT will die consequently but network created by IDT may still be alive when the others still may be able to provide 100%-percent coverage. However, a numerical data of the improvement of IDT over ODT in this case will be an objective of our future simulation.

V. CONCLUSION

In this work, we deal with the area coverage problem with variable sensing radii in WSN by improving the energy balancing heuristic proposed in [2]. The theorem guarantees the correctness of our algorithm. The extensive simulation results prove the efficiency and the stableness of our algorithm in compared with ODT algorithm.

REFERENCES

- [1] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, IEEE International Conference on Communications ICC, 2001
- [2] J. Wang, S. Medidi, “Energy Efficient Coverage with Variable Sensing Radii in Wireless Sensor Networks”, Wireless and Mobile Computing, Networking and Communications, 2007 - WiMOB 2007, White Plains, NY, USA.

- [3] M. Cardei and J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, Journal of Computer Communications on Sensor Networks, 2004
- [4] M. Cardei, J. Wu, N. Lu, M.O. Pervaiz, Maximum Network Lifetime with Adjustable Range, IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob’05), Aug. 2005.
- [5] M. Cardei, J. Wu, M. Lu, Improving network lifetime using sensors with adjustable sensing ranges, International Journal of Sensor Networks, 2006.
- [6] S. Yang, F. Dai, M. Cardei, J. Wu, F. Patterson, On Connected Multiple Point Coverage in Wireless Sensor Networks, International Journal of Wireless Information Networks, 2006 - Springer
- [7] J. Wu and S. Yang, Coverage and Connectivity in Sensor Networks with Adjustable Ranges, International Workshop on Mobile and Wireless Networking (MWN), Aug. 2004.
- [8] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad, Maximum Lifetime of Sensor Networks with Adjustable Sensing Range, 2nd ACIS International Workshop on Self-assembling Wireless Networks (SAWN 2006), Las Vegas, Nevada, USA, June 19-20, 2006.
- [9] X.Y. Li, G. Calinescu, P.J. Wan, Y. Wang, Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks, IEEE Transactions On Parallel And Distributed Systems, 2003
- [10] R. Zheng, G. He, and X. Liu, Location-free coverage maintenance in wireless sensor networks, Tech. Rep. UH-CS-05-15, Department of Computer Science, University of Houston, July 2005.
- [11] C.F. Huang, Y.C. Tseng, The Coverage Problem in a Wireless Sensor Network, pp 519-528, Mobile Networks and Applications, 2005 - Springer.
- [12] C. T. Vu, S. Gao, W. P. Deshmukh, and Y. Li, Distributed Energy-Efficient Scheduling Approach for k-Coverage in Wireless Sensor Networks, 25th Military Communications Conference 2006 (MILCOM 2006), Washington DC, USA, October 23-25, 2006.