Identifying Bounds on Sensing Coverage Holes in IoT Deployments

Yaser Al Mtawa
School of Computing,
Queen's University
Kingston, Ontario K7L 3N6
Canada
Email: yalmtawa@cs.queensu.ca

Hossam S. Hassanein
School of Computing,
Queen's University
Kingston, Ontario K7L 3N6
Canada
Email: hossam@cs.queensu.ca

Nidal Nasser
College of Engineering
Alfaisal University
Riyadh, 11533
Saudi Arabia
Email: nnasser@alfaisal.edu

Abstract— *Sensing coverage in Wireless Sensor Network (WSN) research has received significant attention. The usage of WSNs within the Internet of Things (IoT) mandates taking IoT characteristics into account when considering sensing coverage. These characteristics include heterogeneity, ultra-large scale, dynamicity, randomness, and multiple ownership.*

This paper provides an analytical study of sensing coverage in IoT where sensing resources (sensors) are: random, mobile or static, belong to different owners, and which are heterogeneous in terms of sensing and communication capabilities. We utilize Delaunay Triangulation (DT) to partition the target sensing region into triangles. The vertices of these triangles are IoT sensors. Since intra-triangle coverage holes are not uniform, our goal is to locally detect each hole and provide its bounds. First we determine the existence of an intra-triangle coverage hole, and then we provide a computation of lower and upper bounds of each local coverage hole. Our results are promising, and can be utilized in a multiplicity of coverage applications regardless of the sensors or deployment types.

Keywords— Wireless Sensor Network, Internet of Things, Random Deployment, Coverage, Intra-triangle Coverage, Delaunay Triangulation.

I. INTRODUCTION

The vision behind IoT is to construct a large-scale, yet coherent, unified framework from different building technology blocks. Having one umbrella for different enabling technologies such as WSNs, Radio Frequency Identification (RFID) and 4G networks will utilize the full benefits of these technologies. Integrating such enabling technologies is still the greatest obstacle facing IoT realization. The challenge here is not tailoring enabling technologies such as WSN to adapt for IoT, but rather merge the WSN resources in a larger IoT pool of resources [1]. For example, consider heterogeneous temperature sensors that are deployed in a region and belong to three different sensing providers as shown in Fig. 1. These collective sensors can be viewed as shared resources and their cooperation can provide better quality of service. Usage of such shared resources can be further enhanced with participatory sensors (say in smartphones).

A main challenge in such IoT setting is determining sensing coverage, and determining the existence of coverage holes. Coverage holes exist when there are some points in the sensing field that are not covered by any sensors. Applications vary in their sensing coverage requirements. Some of them require single-sensing coverage, which means any point in the target region should be monitored by at least one sensor; while other applications require high coverage and, hence, require "k" sensors to monitor each point in the region [2].

Existing work on sensing coverage in WSN assume sensors are homogeneous and belong to only one sensing service provider. Most of the research addresses deterministic sensor placement and deployment planning to achieve greater coverage and/or to extend the network lifetime [4, 5]. Our research, on the other hand, investigates IoT sensing coverage where sensors are: a) heterogeneous as they have different functionalities and capabilities, b) randomly deployed which is the norm in IoT, and c) belong to different sensing service providers.

We identify the coverage holes and provide upper and lower bounds for these coverage holes. We utilize Delaunay Triangulation (DT) to partition the target sensing region into triangles. The vertices of these triangles are IoT sensors. Since intra-triangle coverage holes are not uniform, our goal is to locally detect each hole and provide its bounds. Intra-triangle coverage (ITC) procedure is distributed and requires only the vertices of each triangle to involve in the calculation which makes ITC procedure scalable and efficient in terms of power consumption. We provide theoretical analysis of IoT sensing coverage holes, and develop an efficient algorithm to detect coverage holes. We then provide upper and lower bounds of the identified coverage holes, and test the validity of the bounds empirically.

This research contributes towards the realization of a sensing cooperative IoT, in which the available sensing resources (from overlapped and overlaid sensors) are used to achieve a given coverage. As such, applications can determine whether to tolerate some coverage holes or whether to initiate a healing procedure to mitigate some coverage holes in the network. To the best of our knowledge, this is the only research that investigates IoT sensing coverage: identifying the coverage holes locally, and providing upper and lower bounds on each sensing coverage holes.

The remainder of this paper is organized as follows. Section II presents the problem formulation, assumptions of our research. Preliminarily results using Voronoi Diagram (VD) and DT toward efficient coverage will be addressed in Section III. Section IV is devoted to study in detail the Intra-triangle coverage, and the algorithm to detect and bound coverage holes. Section V presents experimental results to validate our proposed algorithm. Section VI concludes the paper and discusses future research directions.

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

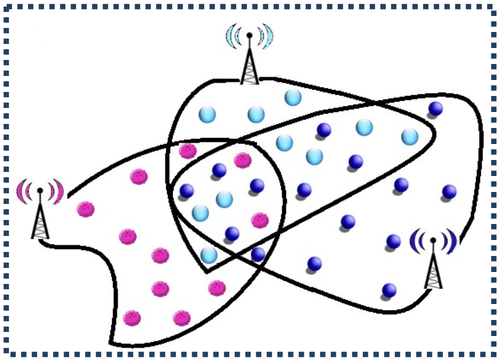

Fig. 1 Multiple IoT sensing providers.

II. Problem description

Given the dynamic and random deployment of IoT sensors, we are interested in detecting the coverage holes and providing upper and lower bounds on coverage holes in a distributed manner. The analysis exploits powerful structures in computational geometry such as the VD and DT. Our approach to detect and bound coverage holes depends only on the locality of each convex polygon of the computational structure that represents the sensing field.

A. Assumptions

We make the following assumptions:

1. IoT sensors can send/receive packets to/from their neighbors. This assumption is important to exchange the IoT sensors' information locally through, probably, multi-hop in order to build our computational structure in a distributed way.
2. IoT sensors know their location.
3. No three IoT sensors are collinear. This assumption enables constructing the Delaunay Triangulation.
4. The sensing target field is bounded. This is the case for most IoT applications.

B. Network Model

Random deployment is the norm in IoT. This refers to the case where the placement of groups of IoT sensors (belonging to different providers) in the target field is independent from other groups. Let $N = \{s_i : s_i \text{ is an IoT sensor}; 1 \leq i \leq n\}$, be the collective set of IoT sensors with location of sensor s_i being (x_i, y_i). Let s_i be an IoT sensor. We define $NH(s_i)$ to denote the neighborhood of s_i, that is, $NH(s_i) = \{s_j | d(s_i, s_j) \leq R_{t,s_i} \text{ and } d(s_i, s_j) \leq R_{t,s_j}, s_i \neq s_j, s_j \in N\}$, where $d(s_i, s_j)$ is the Euclidean distance between s_i and s_j, and R_{t,s_k} is the transmission range of IoT s_k. Each IoT sensor that receives this information is able to estimate its distance from the emitting sensor.

C. Sensing Model

There are different types of sensing models such as binary disk model and probabilistic sensing model. The latter depends on uncertainty in sensor detection. Therefore, it utilizes the detection probability when the point of sensing field is at distance greater than the value of uncertainty, but within the sensing range. In the binary disk model, a point in sensing field is covered if it is within the sensing range of at least one sensor. Otherwise, it is not covered, as given in the following equation:

$$C(p) = \begin{cases} 1 & if\ d(p, s_i) \leq R_{si} \\ 0 & otherwise \end{cases}$$

where R_{si} is the sensing range of IoT sensor s_i. Thus the binary disk model abstracts the sensing coverage of an IoT sensor s_i by a disk of radius R_{si}. In this research, we adopt the binary disk model. Fig.2 shows the disk model representation of the overlapped region in Fig. 1.

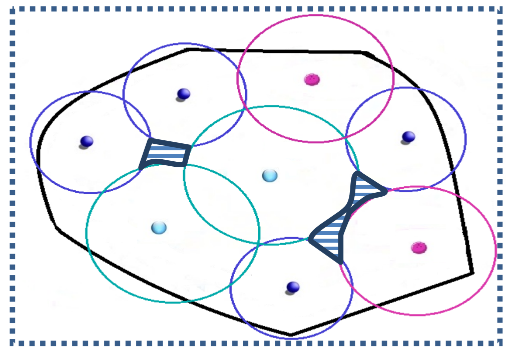

Fig. 2 Disk model of overlapped IoT sensing providers with coverage holes

III. Twoards Efficient Sensing Coverage : Prelimenary Results

We now investigate the full coverage of target sensing field *S*. Initially, we address the randomness of IoT sensors, but with similar values of sensing range, denoted as R_s. We use the following definition of the coverage problem:

Definition 1: Let *S* denote the target sensing field. Let $N = \{s_i : s_i \text{ is a IoT sensor}; 1 \leq i \leq n\}$, be a set of IoT sensors with location (x_i, y_i) in a plane. Each unknown IoT sensor s_i has estimated location (x_i', y_i') and a sensing range R_s. Let *p* be a point in *S*, then *p* is said to be covered if there is at least one IoT sensor s_i such that *p* is within the distance R_s from s_i. In other words, $\{\exists\, s_i | d(p, s_i) \leq R_s, 1 \leq i \leq n\}$.

Let *p* be a point in *S*. We call s_i a dominant sensor of point *p* if s_i has the shortest distance to *p* among all other sensors in *S*. That is $dom(p) = \{s_i | d(p, s_i) = \text{Min}(d(p, s_j), 1 \leq j \leq n)\}$, where $s_i, s_j \in S$. Let $\text{MaxMin}(S) = \underset{p_j \in A}{\text{Max}} \left(d\left(p_j, dom(p_j)\right), j \in [1, \infty)\right), p_j \in S$. Note that if $\text{MaxMin(S)} \leq R_s$ then *S* is fully covered. However, it is not feasible to obtain MaxMin(S) among infinite number of points p_j in *S*. To overcome this problem, we utilize VD to cluster the sensing field *S* into adjacent convex polygons, called cells and denoted by Vor(s_1), Vor(s_2), ….., Vor(s_n). Each cell Vor(s_i) is associated with only one sensor s_i, $1 \leq i \leq n$ as shown in Fig 3. VD is constructed by drawing the perpendicular bisector of the line segment of each pair of sensors. The bisector line segments form the boundary of Voronoi cells. Each bisector line segment is called an edge and the endpoints of this edge are called vertices. For any point p in $Vor(s_i), 1 \le i \le n$, then s_i the closest sensor to p. Note that if p is on a common edge of two neighbouring polygons, then it is equidistant from the two sensors associated with these polygons [12]. The following lemma provides the necessary and sufficient conditions to have full coverage in VD.

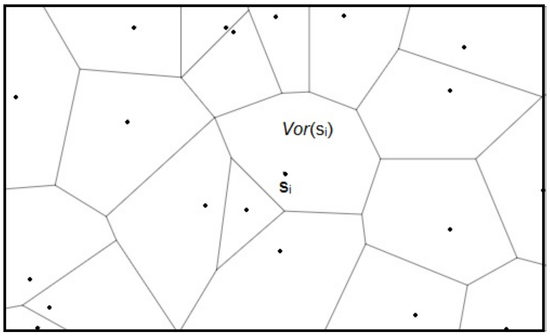

Fig. 3 VD partitions sensing field into convex cells

Lemma 1:

Sensing filed S is fully covered if and only if all vertices in its corresponding Voronoi diagram have a distance less or equal to R_s to their associated sensors.

Proof.

Proving the "only if" part is straight forward. We focus on the "if" part. Voronoi diagram partitions a sensing field into convex cells. The farthest point in any convex polygon from its associated sensor is one of its vertices. Let u be a vertex in Vor(s) with

$$d(u,s) = \operatorname*{Max}_{\substack{j \in [1,n] \\ k \in [1, |V(\mathrm{Vor}(s_j))|]}} d(v_k, s_j), v_k \in V(\mathrm{Vor}(s_j)) \quad (1)$$

and $V(\mathrm{Vor}(s_j))$[1] is the set of all vertices in $\mathrm{Vor}(s_j)$. This means $\mathrm{MaxMin(S)} = d(u,s)$. Therefore, for any point $x \in Vor(s_i)$, $d(x, s_i) \le \mathrm{MaxMin(S)} \le R_s$. □

The coverage problem of sensing field S is now converted, by Lemma 1, from checking non-finite set of points in S into testing a finite set[2] of points that represent the cell's vertices of VD. This lowers the computational cost, adding to the feasibility of the solution. If we could maintain $\mathrm{MaxMin(S)} \le R_s$, we guarantee the full coverage of S. Therefore, VD is a powerful tool to show the existence of coverage holes rather than to quantify the coverage of WSN [13]. This is because Voronoi polygons have different convex shapes with various numbers of edges and have a non-unit-circular model. Therefore, VD does not provide much information about the location and the size of each coverage hole in the field. Thus, we need to have a more efficient structure to control and track the boundary of each coverage hole. That is why we triangulate each Voronoi cell. The vertices of triangulation are the sensors. Two sensors s_i and s_j form a triangle edge if Vor(s_i) and Vor(s_j) have a common Voronoi edge e. This implies that the triangle edge $s_i s_j$ is a segment of the perpendicular bisector line of e. This triangulation is called DT which provides angle-optimal planar triangles such that the circle that circumscribes any triangle, with non-collinear sensors, is devoid of any other sensors. Note that the strong property of convexity in VD is still held in DT as any triangle is the basic convex polygon. Next, we provide a corollary that links the coverage problem to the edges of DT.

Corollary 1:

Let $R_t >= 2R_s$, if the sensing field is fully covered, then the length of every Delaunay triangulation edge is at most $2R_s$.

Proof.

Assume the sensing field S is fully covered, then every point in the field has a maximum distance of R_s from at least one sensor. Let x be a point on the bisector line of segment line joining two sensors s_1 and s_2. Point x is part of the common Voronoi edge between the Voronoi cells associated with s_1 and s_2. This means d(x, s_1) >= d(s_1, s_2)/2 and d(x, s_2) >= d(s_1, s_2)/2. But d(x, s_1) = d(x, s_2) <=R_s which implies that d(x, s_1) + d(x, s_2) <=2 R_s. Thus, d(s_1, s_2) < = 2 R_s. □

A. Variable Sensing Range

The heterogeneity in IoT means that sensors have different sensing ranges. Let R_{si} denotes the sensing range of IoT sensor s_i. Let s_i and s_j are two triangle vertices that have sensing ranges Rsi and Rsj, respectively, then the following lemma holds.

Lemma 2:

The triangle Δ is fully covered if every pair s_i and s_j of its set of vertices satisfies the following conditions:

a) $2r \le R_{si} + R_{sj}$.
b) $d(s_i, s_j) \le R_{si} + R_{sj}$.

Proof. Directly follows from the proofs of Lemma 1 and Corollary 1. □

Note that if L is the longest triangle side in Δ and θ is an angle opposite L (largest angle) then for Lemma 2, it is enough to check condition b) against L only. Furthermore, the conditions in a) and b) are not independent as the relation between r and L is given by $\frac{r}{L} = \frac{1}{\delta}$ where $\delta = 2sin\theta$. This implies that r is not always less than L and, hence, if the circumcenter is inside a triangle Δ, then it is enough, in this case, to check only condition a).

In the next section, we investigate how to detect and define the bounds of each uncovered area in DT.

IV. Intra-Triangle Coverage

The coverage problem is reduced by DT to studying the coverage of each individual triangle in DT. According to Lemma 2, if we could find two vertices s_i and s_j in Δ with $R_{si} + R_{sj}$ less than the maximum of their edge's length and the radius of the circumcircle (the *circumradius*), then there is an uncovered area in the circumcircle of Δ. According to the

[1] The average number of vertices in any Voronoi cell is less than 6 [17].
[2] The size of this set is at most 2n-5 [17].

largest angle θ in Δ, we differentiate three scenarios of the circumcenter: the circumcenter is inside Δ if $\theta < \frac{\pi}{2}$, outside Δ if $\theta > \frac{\pi}{2}$, or on the longest side opposite to $\theta = \frac{\pi}{2}$. Although DT provides the best possible optimal-angle planar triangles (angles around $\frac{\pi}{3}$), in random deployment it is possible to find largest angles greater than $\frac{\pi}{3}$. The following question then arises: What is the minimum density of sensors such that DT is well behaved?

Let R_s be the minimum sensing range among all IoT sensors that participate in the coverage of the target field. In optimal cases, all angles of Δ are equal to $\frac{\pi}{3}$ (i.e. equilateral triangle) and the length of triangle's side is $d = \sqrt{3}R_s$ [8]. Thus the area of a triangle is $\frac{3\sqrt{3}}{4}R_s{}^2$. Furthermore, the number of triangles in any triangulation is *2N-2-k*, where *N* is the number of sensors and *k* is the number of which are on the convex hull of *N* [12]. Assume the sensing field *S* has a size *L x L*; the area size that should be covered by each triangle is $\frac{L^2}{2N-2-k}$. Therefore,

$$\frac{L^2}{2N-2-k} < \frac{3\sqrt{3}}{4}R_s{}^2 \tag{2}$$

This gives

$$\frac{2L^2}{3\sqrt{3}R_s{}^2} + \frac{2+k}{2} < N \tag{3}$$

We assume that the minimum density is achieved. Let s_i be an IoT sensor vertex in a triangle Δ. Then s_i contributes to the intra-triangle sensing coverage of Δ. The coverage contribution of s_i is the size of the angular sector centered at s_i with radius R_{si}. Calculating the contribution of s_i in Δ requires the angle at s_i. Since the lengths of all edges of Δ are known, we use the cosine formula to extract the angle at each sensor. That is

$$\alpha = cos^{-1}\left(\frac{a^2+b^2-c^2}{2ab}\right) \tag{4}$$

where *a, b* and *c* are the lengths of Δ's sides, and α is the angle opposite to side of length *c*. Therefore, the coverage contribution of IoT sensor s_i is

$$CNT(s_i, \Delta) = \frac{\alpha}{2}R_{si}{}^2 \tag{5}$$

where α is the angle at s_i in triangle Δ. The following formula gives the intra-triangle coverage of Δ, denoted by $ITC(\Delta)$. That is

$$ITC(\Delta) = \sum_{s_i \in V(\Delta)} CNT(s_i, \Delta) - A_{mut}, \tag{6}$$

where $A_{mut} = (A_{1,2} + A_{1,3} + A_{2,3}) - A_{1,2,3}$, $V(\Delta)$ is the set of the three vertices of Δ, and $A_{i,j}$ is the common area size contributed by both angular sectors centered at vertices s_i and s_j, and $A_{1,2,3}$ is the area covered by all three vertices.

A. Discovering Coverage Holes

Assume a random deployment of IoT sensors in terrain *S*. Without loss of generality, we assume that IoT sensor network is stationary at the time instance of discovering the coverage holes. A low computational cost and distributed algorithm can be used to construct a DT of *S* such as the localized algorithm in [14]. Clearly, the uncovered area will be inside Δ if the largest angle $< \frac{\pi}{2}$. Otherwise the uncovered area is extended to outside Δ. The latter case will be considered by the intra-coverage analysis of the neighbouring triangle that contains the circumcenter of Δ. However, it is still possible to find part of the uncovered area inside Δ in the case where $\mathrm{d}(\mathrm{s_i}, \mathrm{s_j}) > R_{si} + R_{sj}$ for any two vertices s_i and s_j in Δ. For the first case, the centroid of the coverage hole will be computed as well as the boundary of this hole which will be discussed next.

The intra-triangle uncovered areas have different shapes; however, we model the upper and lower bounds of each uncovered area in triangle Δ as circles. The lower bound circle is a circle centered in the centroid of the polygon that strictly contains the uncovered area in Δ; it is the largest circle that can be inscribed inside the uncovered area. On the other hand, the upper bound circle is the minimum circle that circumscribes the uncovered area of Δ. To compute lower and upper bounds for the uncovered area in Δ, we follow the following procedure: first, we find a set *U* of intersection points, namely the angular sectors and the edges of Δ, and the intersection points of the angular sectors themselves. Let s_i and s_j are two vertices in Δ. If $R_{si} + R_{sj} > d(s_i, s_j)$ we exclude the intersection points between the circles centered in s_i and s_j and the edge s_is_j. Let U' be the new set of intersection points. The points of U' form a polygon *P*. Our goal is to find the minimum/ maximum circle that circumscribes/inscribed-in *P*. To do that, we need to determine the centroid *c* of this polygon. The coordinates of the centroid is given by the following formula [15]:

$$c_x = \frac{1}{6A}\sum_{i=0}^{|U'|-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{7}$$

$$c_y = \frac{1}{6A}\sum_{i=0}^{|U'|-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \tag{8}$$

where *A* is given by $A = \frac{1}{2}\sum_{i=0}^{|U'|-1}(x_i y_{i+1} - x_{i+1} y_i)$, and (x_i, y_i) and (x_{i+1}, y_{i+1}) are two consecutive points on *P*'s hull. Let $R_l = Min_{p_i \in U'}\, d(c, p_i))$. The circle centred in *c* with radius R_l represents a lower bound of the uncovered area in Δ. Likewise, let $R_u = Max_{p_i \in U'}\, d(c, p_i))$. Then $\pi R_u{}^2$ represents the size of the minimum circle that circumscribes *P* and, hence, considers as an upper bound of the uncovered area in Δ.Therefore, we have the following bounding formula:

$\pi R_l{}^2 < Uncovered\ Area < \pi R_u{}^2$ where both bounding circles are cantered at the centroid of a polygon *P* that contains the uncovered area. It should be noted here that the use of centroid *c* instead of circumcenter of Δ is more effective for the following reasons: 1) the circumcenter of Δ does not always belong to the uncovered area due to the variation of IoT sensing ranges. 2) The circumcenter could be outside Δ which makes the calculation of intra-triangle coverage irrelevant. 3) The bounds using the centroid *c* are tighter as it represents the uncovered area more fairly. Next, we present the algorithm that deals with intra-triangle coverage to detect non-uniform uncovered areas and provide a uniform upper and lower bound for these areas.

B. Lower and Upper Bounds (LUB) algorithm

The steps of analytical study to compute lower and upper bounds can be summarized in the following algorithm.

Algorithm 1: *Lower and Upper Bounds (LUB)*

Input: triangle Δ
Output: *c*, *lowerBound*, *upperBound*
If *HasCoverageHole(Δ)* then
    P = *findPolygon(Δ);*
    c = *findCentroid(P);*
    R_l = *findRadiusLowerBound(P, c);*
    R_u = *findRadiusUpperBound(P, c);*
    return *c, lowerBound, upperBound*;
End if

LUB algorithm assumes that all sensors have been localized and their locations are known. While DT is being constructed, each sensor starts to know its neighbours for each triangle Δ in DT. LUB algorithm first check the existence of coverage hole by calling *HasCoverageHole(Δ)* function which simply checks Δ against the coverage criteria in Lemma 2. If a coverage hole is discovered, the function *findPolygon(Δ)* is invoked to find the polygon that strictly circumscribed the uncovered region as discussed in this section. *findCentroid(P)* will apply the equations (7) and (8) to find the centroid of *P*. The remaining is to call *findRaduisLowerBound(P, c)* to calculate the shortest distance between *P*'s vertices and *c* which is the radius of the lower bound. Similiarly, *findRaduisUpperBound(P, c)* returns the longest distance between *P*'s vertices and *c*.

V. NUMERICAL RESULTS AND DISCUSSION

We conduct several experiments where random non-uniform IoT sensors are deployed in the target field. These experiments tend to show the validity of our algorithm and its numerical computation. We use Visual Studio C++ to implement the algorithm. In all experiments, we set the values of the parameters in the following way, unless otherwise stated: the terrain is 300 X 300 m^2, the number of sensors N=350, the average IoT sensing range R_s=15m (with variance of 5 m). The results of all conducted experiments are the average of 10 runs. We use the implementation of a distributed algorithm in [16] to construct the DT that represents the target sensing field. We first study the effect of sensing range on coverage. We also investigate the impact of IoT sensor density on the coverage percentage. Fig. 4 shows that increasing the number of IoT sensors will increase the coverage percentage. The results from Fig. 4 demonstrate the consistency and the validity of our approach in a typical setting with well-understood sensing coverage parameters.

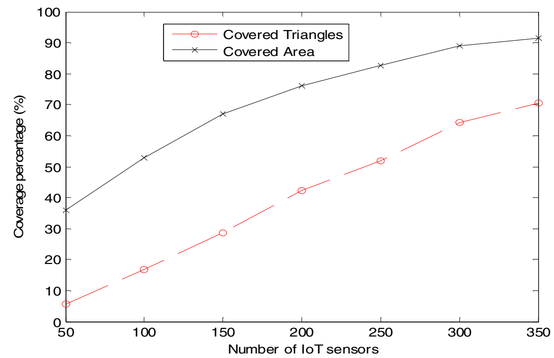

Fig. 4 Coverage percentage vs. sensor density.

We next demonstrate the performance and consistency of the LUB algorithm. To assess scalability, we enlarge the target field to be 500x500 (m^2) with IoT sensors *N*=600, and the value of average sensing range to be 20 m (with variance of 5 m), unless otherwise stated. Fig. 5 shows that the upper and lower bounds are slightly tighter (closer to the actual size of the coverage hole) as the IoT sensor density increases. This is because adding more IoT sensors will fine tune large uncovered areas and, therefore, makes the bounds closer to the actual size of coverage holes. Increasing the density maintains the largest angel of any triangle in DT within the range of optimality (i.e. $\frac{\pi}{3}$), i.e. DT is well behaved. Note that the lower bound is on average twice as much closer to the actual size of coverage hole than the upper bound.

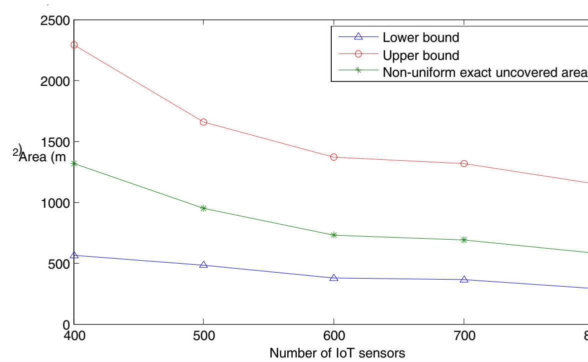

Fig. 5 Lower and upper bound of uncovered area with different sensing range values.

Next, we show the impact of the sensing coverage on the derived bounds. We set the variance of sensing coverage of all IoT sensors to be 5m around the average sensing coverage. The results show that the average IoT sensing range enhances the behaviour of LUB algorithm and the values of the bounds becomes closer to the actual size of the coverage hole as shown in Fig. 6. As the average sensing coverage increases, both bounds converge to the actual size of coverage hole and this is expected as the uncovered areas becomes smaller and finer.

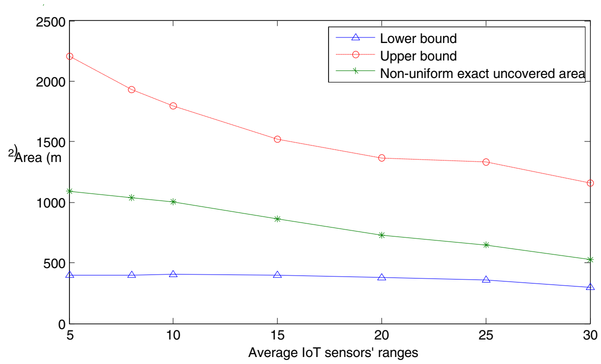

Fig. 6 The impact of average IoT sensing range on LUB.

VI. CONCLUDING REMARKS

The continuing research in IoT sensing coverage is essential for the realization of IoT. This study investigates the IoT sensing coverage problem where hetrogeneous and randomly deplyed IoT sensors are considered. Computational geometry provides a localized approach that enabled us to discover the problem in a distributed manner, by addressing the intra-triangle coverage, detecting the coverage holes, and providing lower and upper bounds for coverge holes. Our findings provide a solid input for many IoT large-scale coverage applications to either tolerate or call a healing procedure to gap, the coverage loss.

The results reinforce the importance of cooperative sesning coverage of multiple sensing providers. Collective IoT sensors not only improve the percentage sensing coverage, but also enhance the identification of the bounds of coverage holes among these networks. This study also shows the possibility that hetrogeneous networks which provide cooperative sensing coverage can expand their lifespan by preserving energy while maintaining the average sensing range at a desired level.

Future work includes the study of possible inaccuracies in localizing some IoT sensors. For example, recent research [18] showed the impact of anchor misplacment on localization accuracy and how this affects the coverage quality. As well, investigating the connectivity of IoT network (i.e. connection coverage) is crucial to keep all IoT resources communicating with each other to exchange information, resource re-use and routing. For instance, relay nodes (RNs) can be deployed to increase the conectivity and lifetime of the network [19]. Another direction is looking into sufficient sensing coverage in overlapped IoT sensors. Given a set of deployed sensors of multiple sensing providers, what is the subset of sensors that provides the best coverage and/or monetary cost.

REFERENCES

[1] S. Oteafy and H. S. Hassanein, "Resource re-use in wireless sensor networks: Realizing a synergetic internet of things," *Journal of Communications,* vol. 7, pp. 484-493, 2012.

[2] J. Liang, M. Liu and X. Kui, "A survey of coverage problems in wireless sensor networks," *Sensors & Transducers (1726-5479),* vol. 163, 2014.

[3] C. Zhu, C. Zheng, L. Shu and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications,* vol. 35, pp. 619-632, 2012.

[4] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications,* 2002, pp. 32-41.

[5] N. Ahmed, S. S. Kanhere and S. Jha, "The holes problem in wireless sensor networks: a survey," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 9, pp. 4-18, 2005.

[6] S. M. Oteafy and H. S. Hassanein, "Re-usable resources in wireless sensor networks: A linear optimization for a novel application overlay paradigm over multiple networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE,* 2011, pp. 1-5.

[7] X. Wang and S. Zhang, "Research on efficient coverage problem of node in wireless sensor networks," in *Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium on,* 2009, pp. 532-536.

[8] G. Takahara, K. Xu and H. Hassanein, "Efficient coverage planning for grid-based wireless sensor networks," in *Communications, 2007. ICC'07. IEEE International Conference on,* 2007, pp. 3522-3526.

[9] G. Takahara, K. Xu and H. Hassanein, "How resilient is grid-based WSN coverage to deployment errors?" in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE,* 2007, pp. 2872-2877.

[10] K. Xu, G. Takahara and H. Hassanein, "On the robustness of grid-based deployment in wireless sensor networks," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing,* 2006, pp. 1183-1188.

[11] H. Kenniche and V. Ravelomananana, "Random geometric graphs as model of wireless sensor networks," in *Computer and Automation Engineering (ICCAE), 2010 the 2nd International Conference on,* 2010, pp. 103-107.

[12] M. De Berg, M. Van Kreveld, M. Overmars and O. C. Schwarzkopf, *Computational Geometry.* Springer, 2000.

[13] H. Chizari, M. Hosseini, T. Poston, S. A. Razak and A. H. Abdullah, "Delaunay triangulation as a new coverage measurement method in wireless sensor network," *Sensors,* vol. 11, pp. 3163-3176, 2011.

[14] M. Li, G. Calinescu, P. Wan and Y. Wang, "Localized delaunay triangulation with application in ad hoc wireless networks," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 14, pp. 1035-1047, 2003.

[15] P. Bourke, "Calculating the area and centroid of a polygon," *URL.Http://local.Wasp.Uwa.Edu.Au/-pbourke/geometry/polyarea,* 1988.

[16] *Delaunay Triangulation Library for C++* (2014). Available: http://www.geom.at/fade2d/html/. [Accessed: 03-Nov-2014]

[17] F. P. Preparata, M. I. Shamos and F. P. Preparata, *Computational Geometry: An Introduction.* Springer-Verlag New York, 1985.

[18] Y. Al Mtawa, N. Nasser , and H. Hassanein, " Mitigating Anchor Misplacement Errors in Wireless Sensor Networks." IWCMC, 2015. In press.

[19] K. Xu, H. Hassanein, G. Takahara, and Q. Wang "Relay node deployment strategies in heterogeneous wireless sensor networks," IEEE Trans. on Mobile Computing, vol. 9, no. 2, pp. 145-159, Feb 2010.