Redundancy detection protocol for area coverage control in heterogeneous wireless sensor networks

Hervé Gokou Diédié[1], Michel Babri[2] and Souleymane Oumtanaga[3]

**[1] Computer Science Department, University Nangui-Abrogoua & INPHB
Abidjan, Côte d'Ivoire**

**[2] Mathematics & Computer Science Department, INPHB
Yamoussoukro, Côte d'Ivoire**

**[3] Mathematics & Computer Science Department, INPHB
Yamoussoukro, Côte d'Ivoire**

Abstract

In a wireless sensor network, eliminating redundancies in area coverage in the deployment zone represents a major challenge where the stake is its lifetime increase. Current works dealing with this topic use different deterministic or probabilistic techniques for nodes redundancies check. In this paper, we propose a quasi-quadratic time distributed algorithm based on a deterministic approach enabling each node to know its exact status even in the context of heterogeneous sensing ranges. Simulations show that this novel technique is more accurate and precise in comparison with other traditional ones.

Keywords: *wireless sensor network, coverage control, redundancy detection, convex hull, ROC analysis.*

1. Introduction

When it is necessary to collect information about a place where human presence is undesirable, difficult or impossible, small electronic devices also called wireless sensors are used; since they are able to process and communicate on demand, some physical quantities they have previously measured [1]. Moreover, the hostile nature of the Region of Interest (RoI) requires to deploy them randomly and especially in large numbers. This high density contributes to the existence of many redundancies in the activities carried out by the nodes including those related to the monitoring of the deployment area. This results in energy losses which shorten nodes lifetime and eventually leads to an early network partitioning; since in such inaccessible environments, the energy of nodes' non-rechargeable batteries has become a scarce resource.

To overcome this situation also known as the coverage problem, one technique consists in making the redundant nodes, once properly identified, turn off their sensing devices and enter into sleep state [2],[3]. This process must be conducted while maintaining the level of coverage required by the application. Works that use this technique and assume that the underlying application does not tolerate any uncovered area, abound in the literature [2], [4]; however, most of them, while relaxing the constraints inherent in this type of networks, base their strategies on unrealistic assumptions e.g. considering that nodes have the same range, or simply use rules that make redundancy check difficult or worse, contribute to the appearance of some uncovered areas (blind points). In this paper, we address the problem of redundancy check from area coverage perspective, considering most of the real-world conditions. We use a geometric approach to determine the area wherein the probability for a node to be redundant, with respect to its neighbors' position, is the highest. The presence in this area of the latter will be used, eventually with additional criteria, to confirm or deny its eligibility.

The remainder of this paper is organized as follows. Section 2, reviews the most relevant works on this topic, collected in the literature. Section 3 describes in detail our redundancies detection protocol. Section 4, discusses the performance evaluations of our contribution and their results. Section 5 draws some conclusions and perspectives about this work.

2. Related work

In wireless sensor networks, target coverage, barrier coverage and area coverage are the three types of coverage problems commonly addressed in the literature. They have attracted many studies in recent years. The last mentioned issue arises when the underlying application does not tolerate any uncovered area in the Region of Interest (RoI). This issue is also presented in the literature as an optimization problem where it is desired that any region of this area is under the supervision of at least k nodes ($k \geq 1$). Its NP-hardness has been demonstrated by authors such as Gupta *et al.* [5] and Yang *et al.* [6]. Methods used to cope with this problem are either deterministic (geometric) or probabilistic. Probabilistic methods are characterized by a reduction of geometric calculations in favor of establishing a relationship between the areas covered by the different nodes likely to be involved in the redundancy check process. Many studies using these approaches exist in the literature. Among the most recent, are the algorithms PBRCA by Wang *et al.* [7] and PSMC by Sihai and Layuan, [8].

In this paper, we essentially focus on deterministic ones among which we find various techniques; the most common are: virtual grid, perimeter coverage, geometric graphs and crossing coverage [9]. Huang and Tseng in [10] were the first to propose the *perimeter coverage* technique. The coverage of a node is here evaluated through the calculation of the central angle of the arc covered by its neighbors. The authors use an inexpensive algorithm for both homogeneous and heterogeneous networks which however fails to apply efficiently its redundancy check process to the internal intersection points. This problem has been successfully solved by Liu *et al.* [11] with their protocol ERPC. But it also seems to be confined only to homogeneous networks. A technique close to perimeter coverage is the concept called *sponsored sector* used by Tian and Georgenas [12] with their protocol Ottawa. The authors propose indeed an off-duty rule wherein a node is eligible, if each sector created in its sensing disk by its neighbors (sponsors) is covered. This contribution is also difficult to apply to heterogeneous networks. Other authors like Bai *et al.*, [13] with GBCPP use a technique based on a *virtual grid* created from several points purposely selected by a node inside its coverage disk. The latter is eligible, if each of these points is covered by at least one neighbor. This method is both complicated and storage expensive because of the relationship to be specified between the grid points and also between each point and each neighboring node. A more efficient and less time consuming solution is proposed by Liu *et al.* [14] with a concept named *the determined distance* used by the VSGCA protocol. This concept was coined to lower the neighbor-to-point distance computation cost and to avoid partial coverage of the grid. However, grid's cells full-coverage is not guaranteed and the performance strongly depends on their dimensions. Moreover, VSGCA algorithm is not explicitly designed for heterogeneous WSNs.

Another technique is to use some well-known geometric figures for the redundancy check process. This is the case of the Voronoi diagram. A major work using this concept is the distributed redundancy detection algorithm by Carbunar *et al*. [15] in which they propose to create around each node a said Voronoi cell. So that two sensors are Voronoi neighbors if they share a Voronoi edge i.e. an edge separating two cells. A node is therefore redundant if all the vertices of the Voronoi diagram created by its two-hop Voronoi neighbors and their intersection points with its own sensing perimeter are covered. This algorithm can be extended to heterogeneous sensor networks but is complicated to implement in practice especially when node density is high. This idea is also echoed by Tsilker [16] when proposing the usage of weighted voronoi diagrams for redundancy check in any heterogeneous wireless networks. Another example is the use of Releaux triangle. Ammari and Das in [17] propose to use six of these triangles to discretize the coverage area for redundancy check process. In this scheme called Het-SSCk, a node with sensing range r is said to be redundant if any of its six overlapping Releaux triangles of width r contains at least k active neighbors. Het-SSCk has a low computational complexity but does not guarantee the coverage of the central area of a node in a heterogeneous context.

The last major technique for redundancy detection is the one called *crossing coverage*. Here, we focus on the intersection points existing between the neighboring nodes' sensing disks. Zhang and Hou [18] propose along with OGDC a localized process which computes the optimal position to be achieved according to the gradually selected nodes' sensing disks intersection points. The selected nodes are those who are closest to the optimal locations while others are put in sleep state. This protocol can minimize the number of active nodes but cannot guarantee minimum coverage hole due to the initialization process and the approximations in nodes selection. Xing *et al*. [19] propose another major sensing disks intersection points check protocol called CCP based on a rule stated in the following theorem.

Theorem 1 *A convex region A is* Ks*-covered by a set of nodes if* (1) *there exist in region A intersection points between nodes or between nodes and A's boundary;* (2) *all intersection points between any nodes are at least* Ks*-covered; and* (3) *all intersections points between any node and A's boundary are at least* Ks*-covered.*

Gallais *et al*. [20] echoed this idea in their localized protocol for redundancy check. But as shown in Figure 1 by Liu *et al.* [21], these conditions are necessary but not sufficient for k-coverage redundancy detection even in homogeneous WSNs. Applying indeed, Theorem1 to topology shown on this figure, node u would be found eligible.

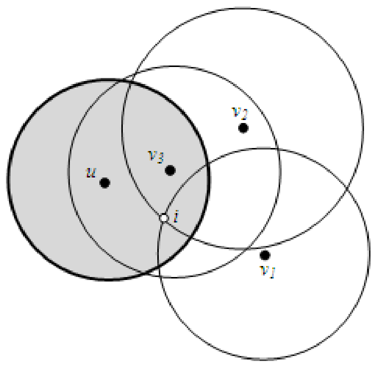

Fig. 1 False redundancy case with the CCP protocol [21].

The purpose of our work is among other things, to provide additional conditions to those described in Theorem 1, for a more accurate and precise redundancies detection process in a context of sensing range heterogeneity.

To conclude this section, let us point that some works use hybrid approaches. Huang and Zhao [22] with the EECRS protocol combine a probabilistic strategy to a deterministic one; respectively based on the calculation of percentage of area covered by neighboring nodes and of the summation of central angles of each arc created by the intersecting disks. A node is eligible if the latter sum equals 360° and the coverage percentage is superior to a defined threshold. Despite a quadratic complexity time, this protocol is not completely distributed because the sink is constantly solicited for the initialization process, with a communication overhead especially for remote nodes.

3. Our Contribution

In this section we give the details of our redundancy check approach. First, we introduce a set of assumptions and some useful definitions that can help the understanding of our strategy.

3.1 Assumptions

We make the following assumptions:

- It is assumed that neighboring nodes intersect both their sensing and communication disks. According to a common principle in the literature (see Xing *et al.* [19]), we consider that for each node x we have $R_x \geq 2 * r_x$ with R_x and r_x respectively denoting its communication and sensing ranges;

- It is also assumed that each node knows its location in the deployment zone by any localization technique like those described in Holger and Wilig, [23] and Mao and Fidan, [24];

-Moreover, it is assumed that the network is heterogeneous, i.e. consisting of nodes with different sensing and communication ranges. This assumption is more than realistic; since nodes ranges depend on their residual energy, non-uniformly consumed by each of them;

- Finally, we assume that the process is taking place in a two-dimensional Euclidian space, even if, theoretically it holds for higher dimensions.

3.2 Definitions

It is important to be acquainted with a few key concepts for a good understanding of our contribution.

Definition 1 (Node Coverage): Node coverage C_u of node u is the region of the deployment zone consisting of all points p under its monitoring. Formally, $C_u = \{p \mid d(u,p) < r_u\}$ where $d(u,p)$ denotes the distance between node u and any point p and r_u node u sensing range.

Definition 2 (Coverage-neighbor): Two nodes u and v are coverage-neighbors, if there are two intersection points between their sensing disks. Formally, knowing that one of the sensing disks is not fully covered by the other, two nodes u and v are coverage-neighbors, if $d(u,v) < r_u + r_v$, where r_u and r_v respectively denote nodes u and v sensing ranges; and $d(u,v)$ the distance between them.

Definition 3 (1-redundancy): 1-redundancy or inclusion-redundancy denotes the situation in which node u coverage is enclosed in another node v sensing disk; as shown in Figure 2(a). Formally, a node u is 1-redundant if $d(u,v) + r_u < r_v$ i.e. u is made redundant by a neighbor v.

Definition 4 (n-redundancy): n-redundancy or combination-redundancy denotes the situation where node u sensing coverage equals the combination of areas it shares with its n coverage-neighbors as shown in Figure 2(b) i.e. u is made redundant simultaneously by n of its neighbors.

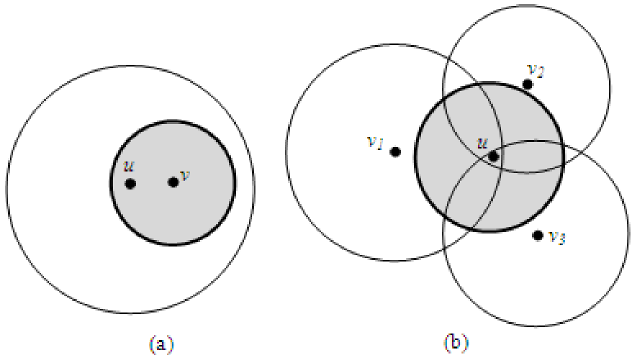

Fig. 2 Two common redundancy cases.

Definition 5 (Maximal Redundancy Zone): the Maximal Redundancy Zone (MRZ) denotes the area where a node needs to be located so as to be n-redundant. The latter zone describes a convex polygon of which hull consists of some of its n coverage-neighbors sensing disks intersection points. As shown in Figure 3, these points are called the *Border Points* while others the *Interior Points*.

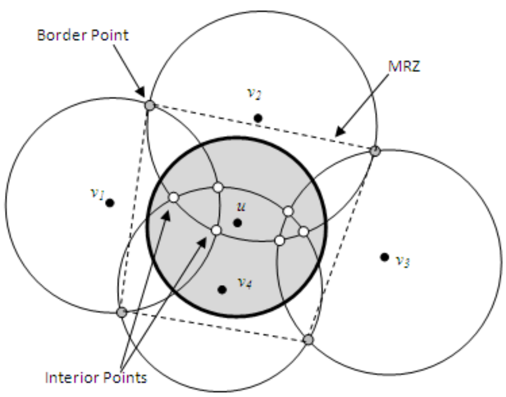

Fig.3 Maximal Redundancy Zone (MRZ) in a 4-redundancy context.

Definition 6 (Candidate): a candidate-node or simply a candidate is a node trying to determine its redundancy status with respect to its neighbors.

Definition 7 (Candidate-Point): a candidate-point is a sensing disk intersection point inside the candidate's coverage.

Definition 8 (Point's parents): parents (father and mother) of a point are nodes of which sensing disks intersection has created this point.

3.3 Redundancy eligibility rules

Rule 1 (1-Redundancy): According to Definition 3, to be 1-redundant with respect to a neighbor v of which sensing range is denoted by r_v, a node u must have a range r_u such as $d(u,v)+r_u<r_v$ with $d(u,v)$ denoting the distance between u and v.

Rule 2 (n-Redundancy): To be n-redundant, a node u with a sensing range r_u must:

- be located inside the Maximal Redundancy Zone (MRZ) defined by its n coverage-neighbors without being 1-redundant for none of them;
- have its sensing range inferior to the distance between its location and each Border point;
- have each of its candidate-points also been covered by at least one neighbor. This requirement is derived from Theorem 1.

The combination of these rules results in the following general algorithm:

Algorithm 1 Redundancy_Check_General

```
Status ← 0 // Node is not redundant
Neighbor Discover y()
if 1-redundant   // Check 1-redundancy
    Status ← 1 // Node is 1-redundant
else            // Check n-redundancy
  Construct the MRZ
  if located inside the MRZ
     if range < Border Points distance
       if Theorem 1 is applied
           Status← 2 // Node is n-redundant
       end if
     end if
  end if
end if

return Status
```

3.4 ARCAD's detailed redundancy check process

ARCAD (ARea Coverage redundancy Accurate Detection) is a localized message passing.protocol. The latter general process will be detailed in the following sections.

3.4.1 Neighbor Discovery

When a node u needs to discover its neighbors, it broadcasts a HELLO probe message, after having emptied its neighbor list. The latter message contains its ID, its sensing and communication ranges respectively denoted by r_u and R_u, including its location coordinates (x_u, y_u). Once such a message is received by a node v, the latter calculates the distance $d(u,v)$ existing between its location and the sending node u, according to the following equation:

$$d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \quad (1)$$

If $r_v > d(u,v)$, node v concludes that the message's sender is a symmetric coverage-neighbor. But if $d(u,v) + r_u < r_v$ node v will conclude that node u is one of its 1-redundants. In either case, node v updates its neighbor list and sends a *WELCOME* message in response. The latter message encapsulates the sender's and receiver's IDs, sender's ranges and location coordinates ; so as the status it has just granted to the *HELLO* message's sender; namely, *Symmetric Coverage-Neighbor* or *1- Redundant*.

After receiving all WELCOME messages, node u updates its neighbor list according to all the information it has collected.

3.4.2 Redundancy check

When receiving a *WELCOME* message, if a node finds out that it has been granted a *1-redundant* status, it considers itself as redundant for the message's sender. Such a neighbor can no more be involved in any n-redundancy relationship with the latter. However, a node can simultaneously be involved in a 1-redundant relationship with several of its neighbors, as shown in Figure 4. This situation represents a multi-redundancy case.

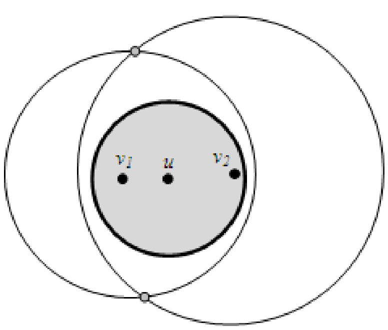

Fig. 4 Multi-redundancy (a double 1-redundancy) case with 2 neighbors.

The n-redundancy check process, as one can guess, is less straightforward. Indeed, when receiving *WELCOME* messages, a node has to build the MRZ from its n symmetric neighbor's coordinates, i.e. those with which it is not involved in any 1-redundancy relationship. It sends to them for that purpose, a *INTERSECT* message asking them to provide information about their mutual relationships, including details about their possible sensing disks intersection points.

After receiving this message, the concerned nodes look for neighbors they have in common with the sender. Then, calculate the coordinates of their sensing disks intersection points, using the following equations:

Let v_1 and v_2 be two neighbors as shown in Figure 5 and let i and j be their sensing disks intersection points of which respective coordinates (x_i, y_i) and (x_j, y_j) must be calculated.

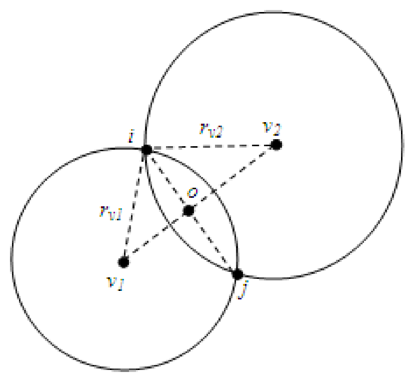

Fig. 5 Two neighbors intersection points.

We have:

$$
\begin{aligned}
x_i &= x_o - \frac{(y_{v2} - y_{v1}) \times \sqrt{r_{v1}^2 - d(v_1, o)^2}}{d(v_1, v_2)} \\
y_i &= y_o + \frac{(x_{v2} - x_{v1}) \times \sqrt{r_{v1}^2 - d(v_1, o)^2}}{d(v_1, v_2)} \\
x_j &= x_o + \frac{(y_{v2} - y_{v1}) \times \sqrt{r_{v1}^2 - d(v_1, o)^2}}{d(v_1, v_2)} \\
y_j &= y_o - \frac{(x_{v2} - x_{v1}) \times \sqrt{r_{v1}^2 - d(v_1, o)^2}}{d(v_1, v_2)}
\end{aligned}
\quad (2)
$$

With:

$$d(v_1, o) = \frac{{r_{v1}}^2 - {r_{v2}}^2 + d(v_1, v_2)^2}{2 \times d(v_1, v_2)} \tag{3}$$

And:

$$x_o = x_{v1} + \frac{(x_{v2} - x_{v1}) \times d(v_1, o)}{d(v_1, v_2)}$$
$$y_o = y_{v1} + \frac{(y_{v2} - y_{v1}) \times d(v_1, o)}{d(v_1, v_2)} \tag{4}$$

These coordinates are included in the *POINTS* message, sent to the requesting node. After receiving this message, the latter node launches the MRZ construction process.

Note that for the 2-redundancy cases, points corresponding to the two neighboring nodes locations will be added to the intersection points list, simply because MRZ needs to have a polygonal shape.

It is advantageous that the point corresponding to the candidate's location also be added to the points required for the MRZ convex hull construction. Such a process will be called the *Relative Maximal Redundancy Zone* (R-MRZ) construction process. This helps to determine in a single operation the limits of the MRZ while evaluating the position of the candidate node with respect to the latter zone (see line 6 and 7 in Algorithm 1). At the end of such a process, the candidate will be located either inside the R-MRZ polygon or on its convex hull.

A candidate located on the R-MRZ convex hull is necessary located outside the MRZ. It can therefore sees itself as a not redundant node for its n neighbors (see line 7 in Algorithm 1); otherwise it must carry on the process by checking that its sensing range is lower than the distance existing between its location and each border point (see line 8 in Algorithm 1), and each candidate-Point is covered according to Theorem 1.

Let us note that when the candidate is not located on R-MRZ convex hull, the latter zone is identical to the MRZ.

The two last steps in the redundancy check process (see line 7-9 in Algorithm 1) can also be combined in a single process. Indeed, to apply Theorem 1, each intersection point can be considered as a candidate-point.

The general method for the n-redundancy detection process (Algorithm 1) can therefore be simplified with two steps. First, construct the R-MRZ, if the candidate is not located on its convex hull then apply Theorem 1.

The R-MRZ construction problem therefore is reduced into finding the convex hull of a cloud of n points; a well-known problem in computational geometry. Graham's scan, Jarvis' march or any similar efficient algorithm can be used to solve this problem.

Algorithm 2 Redundancy_Check_Simplified

```
Status ← 0 // Node is not redundant
n ← Neighbour _Discovery ()
if (n=1) // 1-redundancy check
   Status ← Check_1-redundancy()
else    // n-redundancy check
  if (n=2)
     Points [ ] ← Intersections + Candidate's& Neighbours'Positions
      Ok ← Build_R-MRZ(Points [ ])
      d ← Distance (Candidate, Border Points)
     if (Ok and range < d)
        Status ← 1   // Node is 2-redundant
      end if
  end if
  if (n>2)
      Points[ ] ← Intersections + Candidate 's Position
      Ok ← Build_R-MRZ(Points [ ])
      if (Ok and Apply_Theorem1 (Points[ ]))
        Status ← 1 // Node is n-redundant
      end if
    end if
  end if

return Status
```

Algorithm 2 describes the simplified version of the redundancy detection process as actually used by ARCAD.

Theorem 1 application is to ensure that each candidate-point is covered by at least one of the neighbors involved in the n-redundancy relationship. This process is described by Algorithm 3. As illustrated in Figure 6, a triangulation of the R-RMZ shows that each candidate-point belongs to a triangle, and therefore is likely to be covered by at least one of the parents of the triangle vertices. Each triangle vertex (except the point representing the position of the candidate) is actually a border point of the R-RMZ.

Algorithm 3 Apply_Theorem1 (Points[ ])

```
Verdict ← true
for each Point i in Points[ ]
 for each Neighbour v
     d(v,i) ← sqrt((x_v − x_i)^2 + (y_v − y_i)^2)
    if i is covered and d(v,i) ≤ r_v
       Verdict ← false
       return Verdict
    end if
 end for
end for
return Verdict
```

For $n^2 - n$ points, one can notice that Algorithm 3 has a $O(n^3)$ worst-case complexity time as shown by Xing *et al.* [19]. This is very unfortunate, especially when the number of neighbors is high. Hence, we need to work towards reducing this complexity without compromising the original approach. We propose for that purpose an heuristic described by Algorithm 4.

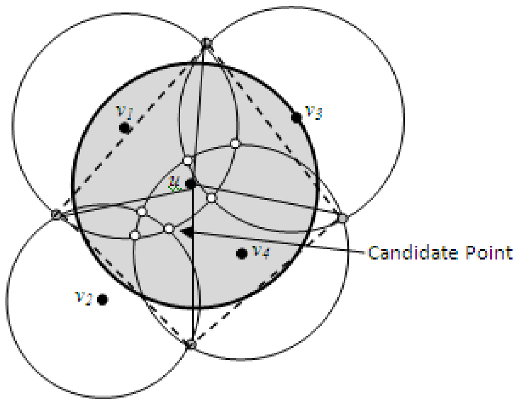

Fig. 6 R-RMZ polygon triangulation.

The rationale behind this heuristic is to take advantage of the relationships that may exist firstly, between candidate-points and border points and secondly, between the different neighboring candidate-points.

Formally, it is to check for a couple of neighboring points if one is covered by either a parent or the *tutor* of the other. A tutor is a node that covers an intersection point.

To do this, each point $p(x, y, f, m, cov)$ must keep respectively in addition to its coordinates, its parents' and tutors' IDs. The process is described in Algorithm 4.

Algorithm 4 Apply_Relation (p1,p2)

```
if (p1 ≠ p2 )

  if (p1.cov=0 and not parent(p2.f,p1)  and   covers (p2.f,p1))
     p1.cov ← p2.f
  end if
  if (p1.cov=0 and not parent(p2.m,p1) and covers (p2.m,p1))
     p1.cov ← p2.m
  end if
  if (p2.cov=0 and not parent(p1.f,p2) and covers (p1.f,p2))
     p2.cov ← p1.f
  end if
  if (p2.cov=0 and not parent(p1.m,p2) and covers (p1.m,p2))
     p2.cov ← p1.m
  end if
  if (p1.cov=0 and covers (p2.cov,p1))
     p1.cov ← p2.cov
  end if
  if (p2.cov=0 and covers (p1.cov,p2))
     p2.cov=p1.cov
  end if
end if
end if
```

For k-coverage redundancy detection, we just have to check that each candidate-point is covered by at least k neighbors. Therefore, Algorithm 4 will need to be slightly modified. Conditions like *p1.cov=0* will be removed, while the number of nodes covering each point should be stored in a variable created for this purpose and attached to each candidate-point.

Theorem 1 application heuristic is described in Algorithm 5. It begins with a descending sort of intersection points according to their abscissa then their ordinates. When scanning the resulting table, information concerning the last three border points (bp_1, bp_2, bp_3) are kept. They will be used to determine the current candidate-point's tutors. Neighboring candidate-points (up to 3 hops) may also be used. The process ends after verifying that each candidate-point is covered by at least one neighbor, as required by Theorem 1.

Algorithm 5 Apply_Theorem1(Points[ ])

```
Descending_Sort (Points[]) // According to abscissa then ordinate
n ← length(Points[ ])
bp1← Points[1]
Ok← false
for  i ← 1 to n
  if (Points[i] on the hull )
    if (bp2≠null)
      if (bp3≠null)
        bp1 ← bp2
        bp2 ← bp3
        bp3 ← Points[i]
      else
        bp3 ← Points[i]
      end if
    else
      bp2 ← Points[i]
    end if
  end if

 Apply_Relation (bp1,Points[i])
 if (bp2≠null) Apply_Relation (bp2,Points[i])
 if (bp3≠null) Apply_Relation (bp3,Points[i])

 if (i+1≤ n)
Apply_Relation (Points[i]),Points[i+1])
 if (i+2≤ n)
Apply_Relation (Points[i]),Points[i+2])
 if (i+3≤ n)
Apply_Relation (Points[i]),Points[i+3])

end for

if (Theorem1 applied) Ok ← true

return Ok
```

4. Performance Evaluation

To validate our contribution let us analyze the time and message complexity of our algorithms and then conduct a series of simulations of which results will be shown and discussed in the following sections.

4.1 ARCAD message complexity analysis

Theorem 2 *ARCAD has a $O(n)$ worst-case message complexity. Where n denotes the number of the candidate's neighbors.*

Proof: ARCAD needs two broadcast messages (*HELLO*, *INTERSECT*) sent by the candidate respectively, for neighbor discovery and intersection points gathering; then two response messages (*WELCOME, POINTS*) sent by each neighbor, totaling $O(n)$ messages. □

4.2 ARCAD time complexity analysis

Theorem 3 *ARCAD has a $O(n^2 \ln n)$ worst-case time complexity. With n denoted by the number of candidate's neighbors.*

Proof: ARCAD requires at most $n^2 - n$ intersection points. The MRZ construction process (finding the convex hull of a point cloud in a 2-D Eucludian space) requires, according to the best known algorithms, a $O(n^2 \ln n)$ worst case upper bounded complexity time. Theorem 1 application heuristic also requires a $O(n^2 \ln n)$ worst-case time for the descending sort and c ($c \le 9$) verifications per point during the points table scanning process, yielding a $O(n^2)$ time.

On the aggregate we have $O(n^2 \ln n) + O(n^2 \ln n) + O(n^2)$. Hence, the overall computational complexity for ARCAD is $O(n^2 \ln n)$. □

4.3 Simulations

In this section, we describe experiments used to validate our contribution. They were conducted using OMNeT++ simulator version 4.5 [25]. The results were compared to those produced under the same conditions by some of the works discussed in section 2.

4.3.1 Experiment I : Accuracy and Precison

This experiment aims to evaluate both accuracy and precision of ARCAD and also to compare these results to those from some of the works discussed above. To do this, we randomly generate different cases of redundant or non-redundant relationships between a node and its neighbors. The number of neighbors (2 to 100) and the nodes ranges are also varied randomly (40 to 80 m). This experiment is conducted in order to perform a ROC (Receiver Operating Characteristic) analysis. The latter is carried out in many fields involving the use of binary classifiers Krzanowski and Hand [26]. Indeed, a redundancy detection protocol can be viewed as such a classifier since it has to identify correctly any case created by the "gold standard" (the random cases generator). Hence, a case involving a redundancy relationship will be said Positive (P) and Negative (N) otherwise. 20,000 such cases were randomly generated and submitted to each protocol. According to its correctness, the result of the detection process by each tested protocol will be therefore labeled as False Positive (FP), True Positive (TP), False Negative (FN) or True Negative (TN).

The experiment results (Table 1-3) were used to determine for each protocol, the True Positive Rate (TPR), the False Positive Rate (FPR), Accuracy (ACC) and Precision (PREC) respectively, according to the following equations:

$$TPR = \frac{TP}{(TP + FN)} \qquad FPR = \frac{FP}{(FP + TN)} \qquad (5)$$

$$ACC = \frac{(TP + TN)}{(TP + FN + TN + FP)} \qquad PREC = \frac{TP}{(TP + FP)} \qquad (6)$$

Accuracy and Precision have been estimated with an asymptotic Confidence Interval (CI) of 95%.

Table 1 Comparison of True Positive and False Positive Rates.

Protocol	*TP*	*FP*	*FN*	*TN*	*TPR*	*FPR*
CCP	9957	607	197	9239	0.981	0.062
ERPC	9598	877	556	8969	0.945	0.089
VSGCA	9927	500	227	9346	0.978	0.051
ARCAD	9821	310	333	9536	0.967	0.031

Table 2 Comparison of Accuracy results.

Protocol	*ACC*	*Asymptotic IC (95%)*
CCP	0.960	[0.957 – 0.963]
ERPC	0.928	[0.925 – 0.932]
VSGCA	0.964	[0.961 – 0.966]
ARCAD	0.968	[0.965 – 0.970]

Table 3 Comparison of Precision results.

Protocol	*PREC*	*Asymptotic IC (95%)*
CCP	0.943	[0.939 – 0.946]
ERPC	0.916	[0.912 – 0.920]
VSGCA	0.952	[0.949 – 0.955]
ARCAD	0.969	[0.967 – 0.972]

Moreover, in a ROC analysis it is customary to compare in a graph (the ROC Space), the *Specificity* to the *Sensitivity* of each tested classifier. These two metrics are equal respectively to (1-FPR) and TPR. Since we are dealing here with discrete binary classifiers, each one will provide a single point on the graph corresponding to the pair (TPR, 1-FPR) [27]. Results are shown in Figure 6 and resized in Figure 7 for clarity.

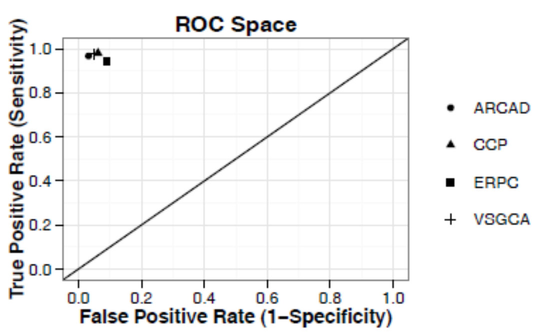

Fig 6 ROC Graph (normal size).

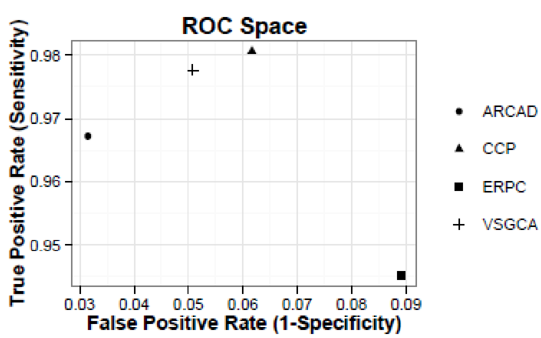

Fig 7 ROC Graph (resized).

4.3.2 Experiment II: Classification quality vs. Cadidate's degree.

The goal of this experiment is to evaluate the influence of the number of candidate's neighbors on the quality of decisions made by ARCAD. The results are compared to those from the protocols mentioned above. The experimental conditions are the same as those described in the preceding experience.

Formally, we have to evaluate the correlation ratio $\eta_{Y|X}$ between a qualitative variable X (the decision) and a quantitative variable Y (the number of neighbors). This indicator is based on the decomposition of the quantitative variable Y variance s_y^2. Therefore, we have:

$$\eta_{Y|X} = \sqrt{\frac{V_{inter}}{s_y^2}} = \sqrt{1 - \frac{V_{intra}}{s_y^2}} \quad (7)$$

$$s_y^2 = V_{inter} + V_{intra} \quad (8)$$

V_{inter} and V_{intra} are respectively, inter-class and intra-class variances.

Each protocol's results are reported in Table 4.

Table 4 Correlation analysis results.

Protocol	V_{intra}	V_{inter}	$\eta_{Y|X}$
CCP	891.10	50.37	0.23
ERPC	874.61	66.87	0.27
VSGCA	900.27	41.20	0.21
ARCAD	928.00	13.52	0.12

Each protocol's variable X conditional characteristics are shown in Figure 8 (a) - (d).

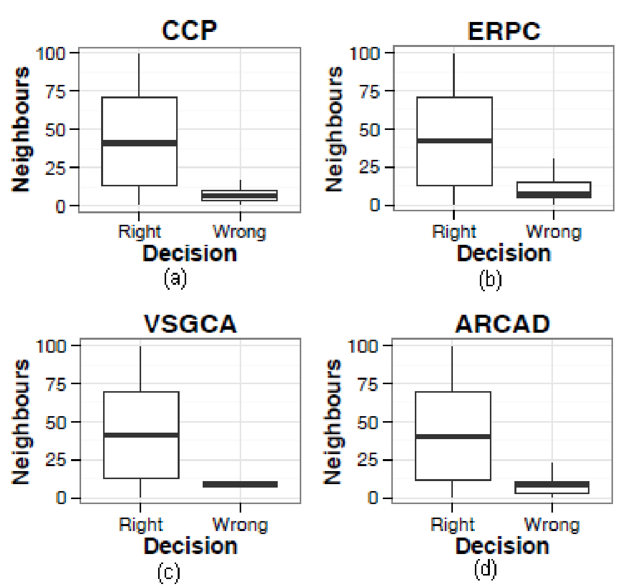

Fig. 8 (a-d) Tukey diagrams of each tested protocol conditional characteristics (without outliers).

4.4 Discussion

In this section we discuss the experiments results presented in the previous section.

4.4.1 Computational complexity

ARCAD has a $O(n^2 \ln n)$ worst-case computational complexity like ERPC but lower than the $O(n^3)$ complexity of CCP with which it still shares the same strategy. This is due to the heuristic used to apply Theorem 1. These complexities however appear to be superior to that of VSGCA ($O(n)$); but the latter strongly depends on the choice of the distance between the points on its virtual grid.

4.4.2 Accuracy and Precision

The results provided by the ROC analysis and illustrated on Figure 6 show that performances of the four tested protocols are well above the random guess line (i.e. diagonal line), and are close to the ideal point on the upper left corner. However, a more detailed analysis of these performances through the ROC Space resized version (Figure 7) shows that CCP has the best True Positive Rate, followed by VSGCA and ARCAD. The latter has by contrast, a False Positive Rate significantly lower than those of other protocols. ARCAD has therefore the best trade-off between True Positives and False Positives. This performance is confirmed by its accuracy and precision, respectively 96.8% and 96.9%, both higher than those of the other protocols (Table 2 & 3). Such performances are due firstly, to the fact that other protocols do not manage explicitly nodes ranges heterogeneity; ERPC is the one that addresses this topic worse, especially with 2 coverage-neighbors (Figure 4). The second reason is the use of insufficient redundancy check rules, as with CCP (Figure 1).

Concerning VSGCA, its virtual grid creation process necessarily produces spaces between points. In a context where nodes ranges are heterogeneous, spaces between points located at the outskirts of the candidate's sensing disk could remain uncovered even if the neighboring points are covered; hence these False Positives cases.

4.4.3 Decision quality vs. Candidate's degree

Evaluation of the correlation ratio between the two variables (Table 4) globally shows that the four tested protocols are weakly affected by the candidate's degree since each ratio is close to 0 ($V_{inter} \ll V_{intra}$) ; albeit with ARCAD this correlation ratio is 0.12. Therefore, ARCAD appears to be the protocol of which decisions are the least influenced by the number of neighbors. This is due to the R-MRZ construction strategy which helps us to quickly determine the existence of a possible redundancy case, unlike ERPC (0.27) and CCP (0.23) which in the context of heterogeneous ranges, are more sensitive to neighbors' position, especially when their number is high.

5. Conclusion and Perspectives

Redundancy elimination in WSNs can contribute to the increase of both nodes and network longevity. However, this policy needs to be conducted while not compromising the deployment area coverage degree required by the underlying application. Hence, the importance of designing reliable redundancy detection protocols with relatively low False Positive Rates. This was the main goal of the work carried out in this paper. To do this, we based our strategy on a geometric method called *crossing coverage* to check the redundancy status of nodes with different ranges. The resulting protocol referred as ARCAD, is based on a localized algorithm whose scalability and efficiency have been proven by the analysis of its computational complexity and the evaluation of its accuracy and precision. They produced respectively a quasi-quadratic time and better results than some of the most relevant solutions in the literature.

However, a scheduling algorithm must be added to such a work, since redundant nodes are intended to be put to sleep. With this additive process we will be able to measure among other things, the impact of ARCAD on the network lifetime.

References

[1] S.S. Iyengar "Introduction to Distributed Sensor Networks" in Mathematical Theories of Distributed Sensor, New-York, USA, Springer-Verlag, 2014 pp.1–.12.
[2] B. Wang "Coverage Problems in Sensor Networks: A Survey" ACM Computing Surveys, Vol. 43, No. 4, 2011, pp. 32:1-32:52.
[3] B. Pazand, A. Datta, and R. Cardell-Oliver "Redundancy Elimination Protocols for Wireless Sensor Networks" in Proc. ICCSA, 2010.,Vol. 6018, pp. 336–351.
[4] G. Fan and S.Y.Jin "Coverage Problem in Wireless Sensor Network: A Survey", Journal of Networks, Vol 5, No 9, 2010, pp. 1033-1040.
[5] H. Gupta, S. Das, and Q. Gu. "Connected Sensor Cover: Self-organization of Sensor Networks" in Proc. ACM Mobile Adhoc Networking and Computing (MobiHoc), 2003, pp.189–200.

[6] S. Yang, F. Dai, M. Cardei, and J. Wu, "On Connected Multiple Point Coverage in Wireless Sensor Networks", Journal of Wireless Information Networks, Vol. 13, No. 4, 2006, pp. 289–301.

[7] Z. Sihai et L. Layuan "Research on Probabilistic Coverage Control Algorithm for Wireless Sensor Network" Advances in Intelligent and Soft Computing, Vol. 116, 2012, pp.853-859.

[8] H. Wang, J. Li, L. Yan, X. Zhang, H. Chen "A Probabilistic based Redundancy Control Algorithm for Wireless Sensor Network," Applied Mechanics and Materials, Vols. 347-350, 2013, pp. 718-723.

[9] B. Wang "Preserving Complete Area Coverage " in Coverage Control in Sensor Networks, London, UK, Springer-Verlag, 2010, pp. 123-127.

[10] C. Huang, Y. Tseng "The coverage problem in a wireless sensor network" in Proc. ACM (WSNA), 2003, pp. 115-121.

[11] Y. Liu, J. Pu, S. Zhang, Y. Liu and Z. Xiong "A Localized Coverage Preserving Protocol for Wireless Sensor Networks," Sensors, Vol. 9, 2009, pp. 281–302.

[12] D. Tian, N. D. Georgenas "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks" in Proc. ACM (WSNA), 2002, pp. 32-41.

[13] H. Bai, X .Chen, and X. Guan, "Coverage for Wireless Sensor Networks of Nodes with Various Sensing Ranges", in Proc. IEEE ICNSC, 2006, pp. 54-59.

[14] Y. Liu, L. Suo, D. Sun, and A. Wang, "A virtual Square Grid-based Coverage Algorithm of Redundant Node for Wireless Sensor Networks", Journal of Network and Computer Applications , Vol. 36, No. 2, 2013, pp. 811-817.

[15] B. Carbunar, A. Grama, J. Vitek, O. Carbunar, "Redundancy and Coverage Detection in Sensor Networks", ACM Transactions on Sensor Networks (ToSN) Vol. 2**,** No.1, 2006, pp. 94–128.

[16] B. Tsilker, "Coverage Ensuring in Heteregeneous Wireless Networks", in Proc. Reability s Stats.in Trans & Comunication, 2008, pp. 305-312.

[17] H. B. Ammari, S. K. Das, "Scheduling Protocols for Homogeneous and Heterogeneous k-covered Wireless Sensor Networks", Pervasive Mobile Computing, Vol. 7, No. 1, 2011, pp. 79-97.

[18] H. Zhang, J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", Ad Hoc & Sensor Wireless Networks, Vol. 1, 2005, pp. 89–124.

[19] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless and C. Grill, "Integrated Coverage and Connectivity for Energy Conservation in Sensor Networks", ACM Transactions on Sensor Networks (ToSN) Vol. 1**,** No.1, 2005, pp. 36–72,.

[20] A. Gallais, J. Carle, D. Simplot-Ryl and I. Stojmenovic, "Ensuring Area k-Coverage in Wireless Sensor Networks with Realistic Physical Layers" in Proc. IEEE Conf. on Sensors, 2006, pp. 880-883.

[21] Liu, Y., Pu, J., Zhang, S., Liu, Y. and Xiong Z. "A Localized Coverage Preserving Protocol for Wireless Sensor Networks" Sensors, Vol. 9, 2009, pp. 281–302.

[22] S.Huang, X. Zhao, "Redundant Nodes Elimination in Wireless Sensor Networks" Communications in Computer and Information Science Information Computing and Applications, Vol. 391, 2013, pp. 48-58.

[23] K. Holger, A. Wilig, "Localization and Positioning" in Protocols and Architectures for Wireless Sensor Network., London, UK, John Wiley & Sons, 2005, pp. 231-249.

[24] G. Mao, B. Fidan "Introduction to Wireless Sensor Network Localization" in Localization Algorithms and Strategies for Wireless Sensor Network, London, UK, Information Science Reference, 2010, pp. 1-32.

[25] OMNeT++ 4.4 Network Simulation Framework, official web site http://www.omnetpp.org.

[26] W. J. Krzanowski, D. J. Hand "Introduction Chapter" in ROC Curves for Continuous Data, Chapman & Hall/CRC Press, 2009, pp. 1-11.

[27] T. Fawcet, "An Introduction to ROC Analysis", Pattern Recognition Letters, Vol. 27, 2005, pp. 861-874.

Hervé Gokou Diédié is currently a PhD student in Computer Science at University Nangui-Abrogoua (UNA) in Abidjan, Côte d'Ivoire. He received his MS in Computer Science from the same university in 2012. His research interests include distributed algorithms, topology control and self-organization in wireless sensor networks. He is a member of Laboratoire de Recherche en Informatique et Télécoms (LARIT) since 2010.

Michel Babri received his PhD in Computer Science from University Clermont-Ferrand, France in 1995. Since 1987, he is a Professor in Computer Science and researcher at INPHB of Yamoussoukro in Côte d'Ivoire. His research interests include operating systems and distributed systems. He is a member of Laboratoire de Recherche en Informatique et Télécoms (LARIT) since 2006, and currently manages the Laboratoire de Recherche en Informatique (LARIT) at INPHB.

Souleymane Oumtanaga received his PhD in Computer Science from University Paul Sabatier of Toulouse, France in 1995. During 1999 to 2000, he stayed in Laboratoire de Recherche en Informatique et Mathématiques Appliquées (LARIMA) at INPHB (Institut National Polytechnique Houphouet-Boigny), Côte d'Ivoire. Since 2000, he has been the Head of Centre de Formation des Technologies de l'Information et de la Communication (CFTIC) of INPHB. Since 1990, he has also been the Head of the Network Information Center (NIC) of Côte d'Ivoire. He has been a Professor in Computer Science since 2007 he currently manages the Laboratoire de Recherche en Informatique et Télécoms (LARIT) at INPHB. His research interests include IP mobility, IP networks security, IPv6, wireless networks, and mobile networks.is a member of the IEEE and the IEEE Computer Society. Do not specify email address here.