MOBILITY AND SECURITY MODELS FOR WIRELESS SENSOR NETWORKS USING VORONOI TESSELLATIONS

Manel Abdelkader, Mohamed Hamdi and Noureddine Boudriga
Communication Networks and Security Research Lab., University of Carthage, Carthage, Tunisia

Keywords: Wireless sensor networks, Collaborative computing, Voronoi tessellation, Mobility models.

Abstract: Recent advances in integrated electronic devices motivated the use of wireless sensor networks in many applications including target surveillance and tracking. A number of sensor nodes are scattered within a sensitive region to detect the presence of intruders and forward subsequent events to the analysis center(s). Obviously, the sensor deployment should guarantee and optimal event detection rate. This paper proposes a high-level Voronoi-based technique to assess the area coverage based on information available locally for each sensor node. We show that the proposed technique can be used to implement a coverage-preserving mobility process to enhance the initial sensor deployment. We also highlight other potential applications of our approach.

1 INTRODUCTION

Wireless Sensor Networks (WSNs) are among the technologies that will probably shape the first decades of the twenty first century. These networks are mainly cost-effective, easy to deploy, and multi-purpose. In fact, WSNs have been used in various contexts including mobile target detection, healthcare, water resource monitoring, and virtual reality. However, they are also characterized by severe memory, CPU, and (most importantly) energy limitations that hardens their deployment in environments where voluminous data should be processed and high-speed networks are to be used to transmit these data. For instance, existing WSNs devised for military surveillance often provide coarse data about the hostile target(s) moving in the battlefield. This constraint, mainly fixed by the sensor node cost, considerably affects the efficiency of the WSN.

The major WSN design issue that should be considered in target tracking applications is area coverage. This is because a sensor node detects the presence of hostile targets only if they are within its sensing range. Therefore, the WSN detection performance depends on how well the sensors observe the physical space. In (Gui and Mohapatra, 2004), coverage degree has been thought of as a measure of the WSN quality of surveillance. A metric, called Average Linear Uncovered Length (ALUL), has been developed to estimate the average distance a mobile target can make before being detected by the sensor network. Therefore, the ALUL can be used to assess the detection efficiency of the WSN. However, the major shortcoming of this approach is its heavy computational load making it non-conforming with the severe processing and energy limitations characterizing WSNs. This complexity is exacerbated when the metric is extended to k-coverage assessment, where $k > 1$.

This paper proposes a coverage assessment approach amenable to implement advanced target tracking functionalities using a hybrid framework composed of a large number of resource-impoverished sensor nodes and a small number of powerful sensor nodes. We rely on a WSN framework, called WHOMoVeS (Wireless Hybrid Optimal Mobile Vehicle Sensing), which has been introduced in (Obaidat, 2008) for military surveillance. To address this issue, we build a higher-order Voronoi diagram of the monitored region for an efficient estimation of the local coverage degree. A collaborative computing framework is set up to fulfill this task. Throughout the paper, we give a general overview of the WSN tasks that may take benefit of the proposed coverage assessment method (i.e., mobility modeling, activity scheduling). To the best of our knowledge, this is the first time higher-order Voronoi tessellations are used in the WSN context, even though simple Voronoi diagrams have already been investigated (Wang et al., 2007; Stojmenovic et al., 2003; Vieira et al., 2003).

The major contributions of the paper are listed in the following:

- The proposed cooperative coverage assessment approach considerably reduces the computational

Abdelkader M., Hamdi M. and Boudriga N. (2010).
MOBILITY AND SECURITY MODELS FOR WIRELESS SENSOR NETWORKS USING VORONOI TESSELLATIONS.
In *Proceedings of the International Conference on Data Communication Networking and Optical Communication Systems*, pages 104-113
DOI: 10.5220/0002989501040113

complexity with regard to existing methods

- Besides coverage optimization, higher-order Voronoi tessellations can be useful for performing multiple tasks including activity scheduling or distributed cryptographic protocols
- Local coverage information, gathered using the Voronoi diagram, can be used to implement coverage preserving mobility models. A simple model is presented in this paper to show that our idea considerably enhances the WSN target detection performance

The rest of the paper is structured as follows. Section 2 gives an overview on the WHOMoVeS and ALUL concepts, which are of utmost importance in our work. A mathematical introduction to Voronoi tessellations is presented in Section 2. Section 3 highlights the potential applications of k-Voronoi diagrams in WSNs. We particularly show in Section 4 how this tessellation is used to develop a coverage-preserving mobility model. The higher-order Voronoi diagram is also shown to be helpful in defining activity scheduling strategies and security schemes. Finally, Section 8 concludes the paper.

2 RELATED WORK

This section first reviews a WSN framework that has been introduced by the authors in (Obaidat, 2008). Then, we give the basic definitions related to the ALUL metric.

2.1 WHoMoVeS: a Framework for Military Target Tracking

At this stage, the reader may wonder about the choice of WHOMoVeS as a WSN framework. In fact, the main reason motivating this choice is that WHOMoVeS builds upon a heterogeneous multi-layer architecture enabling the support of advanced (image and electromagnetic-based) target tracking functionalities, which is very important for military applications. More accurately, Hamdi et al. (Obaidat, 2008) present WHOMoVeS as an heterogeneous sensor network composed of two layers:

- The *core layer*, consisting of sensor nodes equipped with powerful data gathering and transmission capabilities
- The *sensing layer* consisting of miniature devices whose role is restricted to the detection of hostile presence

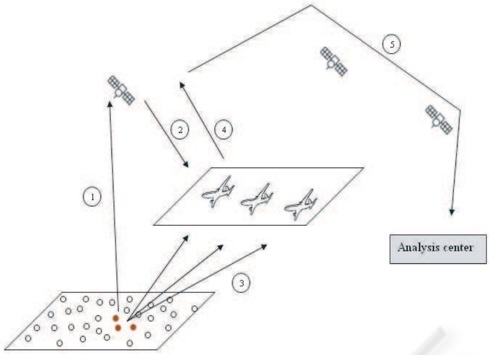

Figure 1: Architecture of the proposed satellite-based communication backbone.

Accurate tracking and long network lifetime are achieved through a strong cooperation between those layers.

According to this reasoning, the process of acquiring and analyzing data related to mobile targets in the battlefield includes five steps as illustrated in Figure 1. These steps are briefly described in the following:

1. Ground sensors detect the presence of a hostile target in the monitored field and store the events in memory. The satellite periodically contacts sensor nodes to download updates about target presence.
2. The satellite contacts the Uninhabited Aerial Vehicles (UAVs) to acquire image data about the scene where the intrusion has been detected.
3. The UAVs gather image data through the embedded imaging sensors.
4. The UAVs establish connections with the satellite communication backbone in order to transmit high-quality multimedia data about the battlefield.
5. Images related to multiple intrusion events are forwarded through the broadband satellite backbone to the analysis center where advanced tracking functionalities are carried out.

2.2 The ALUL Metric

The Average Linear Uncovered Length (ALUL), introduced in (Gui and Mohapatra, 2004), gives an approximation of the distance that can be made by a target before being detected by the sensor network.

The undetected path length of a target traveling from location (x,y) with direction θ is given by the the Linear Uncovered Length (LUL), denoted by $\mathcal{L}((x,y),\theta)$. The average of target paths before detection at location x and over all directions is the Average

Linear Uncovered Length ($ALUL(x,y)$) which means the average distance can be traveled by a target at the location (x,y) without be detected. Obviously, if (x,y) is within the coverage of at least one sensor node then $ALUL(x,y)$ equals 0. More generally, $ALUL(x)$ is calculated as follows:

$$ALUL(x,y) \equiv \begin{cases} 0 & : \text{(x,y) is covered} \\ \frac{\int_0^{2\pi} \mathcal{L}((x,y),\theta)d\theta}{2\pi} & : \text{otherwise} \end{cases} \tag{1}$$

The ALUL in an area A, denoted by $ALUL(A)$ is the mean uncovered distance that can be traveled by a target without being detected by any of the nodes deployed in the region of interest. The expression of $ALUL(A)$ is given by:

$$ALUL(A) \equiv \frac{\int_{(x,y)\in A} ALUL(x,y)dxdy}{\parallel A \parallel}, \tag{2}$$

where $\parallel A \parallel$ is the area of A.

3 HIGHER-ORDER VORONOI TESSELLATIONS AND WSN COVERAGE

The objective of this section is to provide a tool for accurately gauging the coverage degree of the monitored zone. To this purpose, we rely on higher-order Voronoi diagrams to determine the sub-regions that do not satisfy the k-coverage requirement. First, we give a mathematical representation for higher order Voronoi tessellation, which is a set of Voronoi cells. Then, a parallel calculation framework allowing an efficient computation of this tessellation is provided.

3.1 Mathematical Modeling of Higher-order Voronoi Diagrams

We start by the definition of the mathematical model related to sensor nodes distribution. We identify the groups of the k-nearest neighbors using the higher order Voronoi model.

Let $\mathbf{M}$ be a metric space; $\delta : \mathbf{M} \times \mathbf{M} \to \mathbb{R}$ denoting the Euclidean distance on $\mathbf{M}$. We denote by $R = \{p_i, 1 \leq i \leq N\} \subseteq \mathbf{M}$, a set of N sensor nodes having their coordinates in $\mathbf{M}$.

The Voronoi diagram associated to R is the unique subdivision defined in $\mathbf{M}$ such that every part of the subdivision contains the nearest neighbors defined in $\mathbf{M}$ for $p_i, 1 \leq i \leq N$, in R. Every subdivision part is named a Voronoi Cell related to $p_i, 1 \leq i \leq N$, and is determined using the following process.

For every $p_i, p_j \in R$, we denote by $H(p_i, p_j)$ the half plane containing p_i:

$$H(p_i, p_j) = \left\{x \in M/\delta(p_i,x) < \delta(p_j,x)\right\}. \tag{3}$$

It can be noticed that $H(p_i, p_j)$ is the half plane delimited by the bisector of line segment $[p_i, p_j]$ and including p_i.

The Voronoi cell related to p_i is generated by the definition of the common area between all half-planes defined above and containing p_i. Therefore, a Voronoi cell related to p_i is expressed by:

$$V_R(p_i) = \bigcap_{\mathrm{p_j \in R \setminus \{p_i\}}} H(p_{i,} p_j). \tag{4}$$

In our work, we are rather interested in partitioning M into isotopic cells according to k-nearest neighbors for a given distribution of $p_i, p_j \in R$. Starting from a given sensor distribution, we search the set $P_i^{(k)} = \{p_{i1}, \ldots, p_{ik}\}$ containing the nearest sensor neighbors. Such groups can be obtained using the *higher order-k Voronoi Diagram*. The latter allows defining subsets of $\mathbf{M}$ containing the nearest elements to $P_i^{(k)}$. This can be performed by finding the elements which are closer to the most distant neighboring of $P_i^{(k)}$ than any other $p_j \notin P_i^{(k)}$. In other terms, it can be written that:

$$V(P_i^{(k)}) = \left\{ x | \forall p_j \in P \setminus P_i^{(k)}, \max_{p_{it} \in P_i^{(k)}} (\delta(x, p_{it}) \leq \delta(x, p_j) \right\}. \tag{5}$$

$P_i^{(k)}$ is called the generator of this Voronoi cell $V(P_i^{(k)})$. As for the order-1 voronoi cells, an order-k cell is constructed using bisectors between its generators and the remaining of the metric space.

$$V(P_i^{(k)}) = \bigcap_{\mathrm{p_j \in R \setminus P_i^{(k)}}} [H(p_{i1}, p_j) \cap \ldots \cap H(p_{ik}, p_j)].$$

3.2 k-Voronoi Diagram Construction

In this section, we define the k-Voronoi diagram construction model. The latter is based on the cooperation of R elements. In the following, we present the construction k-Voronoi diagram construction algorithm.

Assumptions:

- Every iteration is related to an initial distribution of R's elements on M.
- Every sensor node present in the sensor layer knows his "direct" neighbors (defined in his detection coverage or given by a core sensor).
- The presented algorithm is based on the parallel PRAM algorithm. In this algorithm, we present the parallel construction of k-Voronoi diagram.

Algorithm 1: PRAM Algorithm.

Input:A set R of planar sensors, voronoi of order $k-1$.

Output: the Voronoi diagram of order k.

1. Subdivide each region r_i^{k-1}induced by $P_t^{(k-1)} \subset R$ into subregions according to $V_1(RP_t^{(k-1)})$
2. Merge equivalent new subregions relevant to neighboring r_i^{k-1}.
3. Delete old edges and save the new vertices and edges of each $r_{i'}^{k}$.

4 A K-VORONOI-BASED MOBILITY MODEL

In this section, we show how higher-order Voronoi diagrams can be used to implement sensor mobility modeling. We consider two mobility models:

- The first is an advanced model in which sensor nodes move toward regions where the hostile target is supposed to be
- The second relies on estimating the uncovered zones within a Voronoi cell and moving sensor nodes toward the 'most uncovered region'

4.1 Advanced Mobility Model

Obviously, the first model is more energy-consuming since it encompasses the prediction of the target position. Therefore, we suppose that the second model can be used when energy resources become scarce. The performance of both models will be assessed in the following sections.

Moreover, the prediction function is tightly related to the coverage of the studied zone. In fact, the greater is the number of target detection signals, the better is the prediction precision. In the following, we distinguish both cases of a target crossing a $k-$covered and a non $k-$covered zone.

For a Target Crossing a $k-$covered Zone

The mobility algorithm is triggered upon the detection of a target presence. Every ground sensor sends his detection signal to the relevant intermediate sensor. The latter collects all detection signals, verifies their integrity and defines the zones that might include the target. The set of defined zones are classified according to the probability of presence of the target. This probability reaches his maximum when a zone is $k-$covered. The mobility algorithm is defined as follows:

1. The nearest k sensors s_i, $1 \leq i \leq k$, send their detection signals to their intermediate sensors.
2. In the case where detection signals are sent to different intermediate sensors, the latters coordinate to gather all signals at the IS with the highest number of detection signals.
3. IS verifies the $k-$security of the received signals and constructs the zone of presence of the target z_t.
   (a) Let's $d_i$be the detection signal of the sensor s_i. $d_i = (r_{ti}, \alpha_{ti}, \theta_{ti}, s_i)$ where $r_{ti} = \sqrt{(x_{s_i} - x_{t_i})^2 + (y_{s_i} - y_{t_i})^2}$, $\alpha_{ti} = \tan^{-1}(\frac{y_{s_i}-y_{t_i}}{x_{s_i}-x_{t_i}})$, θ_{ti} is the detection instant. For every s_i, IS computes the detection zone z_i such that $z_i = \int_{\alpha_{ti}-\delta\alpha}^{\alpha_{ti}+\delta\alpha} \int_{r_{ti}-\delta r}^{r_{ti}+\delta r} d\alpha dr$ where $\delta\alpha$, δr are the estimated detection error. The total target presence zone is resulted from the intersection between all the elementary detection zones. Thus, $z_t = \bigcap\limits_{1 \leq i \leq k} z_i$
4. IS defines ΔZ as the zone surrounding z_t and that a target can not go beyond in the next mobility step. IS computes the intersection between $Z_T = z_t + \Delta Z$ and the $k-$Voronoi diagram: $\bigcup\limits_{p_i^{(k)} \in S_k^*} (Z_T \cap V(P_i^{(k)})) = \bigcup\limits_i \delta V_i^{(k)}$, where $\delta V_i^{(k)} \subseteq V_i(P_i^{(k)})$ such that $V_i(P_i^{(k)})$ is the Voronoi cell of the $k-$sensors with index i.
5. To guarantee $k-$coverage in Z_T, each $\delta V_i^{(k)}$ should be $k-$covered which means that $\delta V_i^{(k)} \subset \bigcap\limits_{1 \leq j \leq k} \Gamma(s_j, R_s)$.
6. A mobility instruction is defined by (r_i, α_i) where $r_i \geq d(s_i, p)$ such that $\exists\, p, \forall q \in \delta V_i^{(k)}, d(s_i, p) \geq d(s_i, q)$. and $\alpha_i = argmax\widehat{xs_iy}$ where $x, y \in v_i$ and $v_i$is the set of the vertices of δV_i.

For a Target Crossing a non $k-$covered Zone

In this case, only k' signed detection signals are retrieved by the intermediate sensors. IS proceeds at the construction of the probable zone of presence of the target as presented previously. In the same time, in order to refine the target presence zone, IS starts the recovery of the remaining $(k-k')$ required signals. For this purpose, *IS* proceeds as follows:

1. let's z_i be a probable zone of presence of a target, p_i be the probability of presence of a target where $p_i = k_i/k$ such that $k_i$is the number of the verified

detection signals received by the *IS* and k is the minimum required number of signals.

2. *IS* defines the nearest k sensors to each part of the zone z_i. Thus, *IS* defines the intersection between z_i and the $k-$Voronoi diagram and deduces $\bigcup_i \delta V_i^{(k)}$.
3. For each $\delta V_i^{(k)}$, IS ascertain the sets of the nearest $k$sensors, verifies which sensors k_i", $0 \leq k_i" \leq k'$, have sent detection signals. IS classifies $\delta V_i^{(k)}$ according to the value of k_i". The greater k_i" is, the most important is the probability of presence of the target in $\delta V_i^{(k)}$. A small value of k_j" induces that the target is going in or out $\delta V_j^{(k)}$.
4. For each $\delta V_i^{(k)}$, IS guides the $(k-k")$ nearest sensors to move towards $\delta V_i^{(k)}$. For that, he sends them the mobility instruction including the probability of presence of a target A mobility instruction is defined as.(r_i, α_i, p_i) where $r_i \geq d(s_i, p)$ such that $\exists p, \forall q \in \delta V_i^{(k)}, d(s_i,p) \geq d(s_i,q)$. and $\alpha_i = argmax\widehat{xs_iy}$ where $x, y \in v_i$ and $v_i$is the set of the vertices of δV_i, $p_i = k"/k$ is the probability of presence of the target in $\delta V_i^{(k)}$.

To enhance coverage while keeping more mobility freedom, we suggest a group mobility model in which ground sensors move in groups such that they preserve a $k-$coverage. For this purpose, for each mobility step, sensors define randomly groups of k members for each, the latters are not required to be the nearest neighbors. Each group defines randomly a head which chooses the first mobility step. The remaining members of the group take into account this choice to determine, in turn, their next mobility step. By this manner, each sensor's mobility step depends on his integrating group. Further, a sensor may move from one group to another in each mobility step. This model enables the definition of overlapping $k-$Voronoi groups which increases the guarantee to have a $k-$coverage.

4.2 Simplified Mobility Model

We propose a mobility model which is only based on the Voronoi diagram. The following proposition gives a condition for a Voronoi cell to be partly uncovered.

Proposition 4.1. *Let S be a set of sensor node positions and s_i in S be a sensor node. If there exists n_j in $N(s_i,V(S))$ such that $d(s_i,n_j) > 2R_{s_i}$ then $V(s_i)$ is not fully covered.*

Proof. We suppose that $d(s_i,n_j) > 2R_{s_i}$. Let $[v_p, v_q]$ be the Voronoi edge defined by n_j and s_i. The intersection of $[v_p, v_q]$ and $[s_i, n_j]$ is denoted by P. The properties of the Voronoi diagram give that:

$$d(s_i,P) = d(n_j,P) = \frac{d(s_i,n_j)}{2}. \tag{6}$$

Since $d(s_i,n_j) > 2R_{s_i}$, we deduce from Equation 6 $d(s_i,P) > R_{s_i}$.

Consider the point $Q \in [s_i,P]$ such that $d(s_i,Q) = R_{s_i}$. We can conclude that for every $T \in [P,Q]$, $T \in V(s_i)$ and $T \notin \Gamma(s_i,R_{s_i})$ (because $d(s_i,T) > R_{s_i}$). This means that $V(s_i)$ is not totally covered. □

This result can serve to implement a mobility algorithm where a sensor node looks for one or more neighbors that are at least $2R_{s_i}$-distant from it. If such nodes exist, the sensor node moves toward the most distant neighbor, denoted by n_f, with a distance $\frac{d(s_i,n_f)-2R_{s_i}}{2}$.

Figure 2 illustrates this reasoning. In fact, we notice that the disc centered in s_1 and having a radius equal to R_{s_i} does not cover the Voronoi cell generated by s_1. Hence, s_1 will move toward s_3 with a distance $d(Q,P)$.

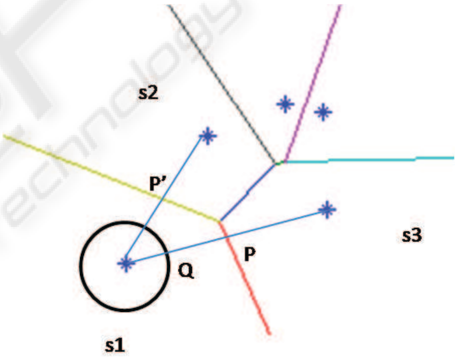

Figure 2: Simplified mobility model.

The following corollaries extend this strategy to the case where the monitored region is required to be k-covered. For the sake of parsimony, we do not provide proofs for these corollaries in this paper.

Corollary 4.2. *For s_i in S, if $|N(s_i,V(S))| < k$, where $|.|$ denotes set cardinality, then $V(s_i)$ is not k-covered.*

Before giving the second corollary, we define, for a sensor node s_i in S, the set $X(s_i,V(S))$ of intersection points expressed as follows:

$$X(s_i,V(S)) = V\widetilde{(S \setminus \{s_i\})} \bigcap \Gamma(s_i,R_{s_i}), \tag{7}$$

where $\widetilde{P}$, for $P \in \mathbb{R}^2$ denotes the boundary of P.

Informally speaking, $X(s_i,V(S))$ denotes the intersection of edges of the Voronoi diagram $V(S \setminus \{s_i\})$ and the disk corresponding to the maximum sensing coverage range of s_i.

Corollary 4.3. *For s_i in S, if $|X(s_i, V(S))| < k$, then $V(s_i)$ is not k-covered.*

The major advantages of these results is that we can rely on simple Voronoi diagrams to deal with *k*-coverage while the advanced model proposed in the previous subsection is based on *k*-Voronoi tessellations which are more complex to build. A more accurate comparison between the two models will be carried out in the simulation section.

4.3 Group Mobility Modeling

To enhance coverage while keeping more mobility freedom, we suggest a group mobility model in which ground sensors move in groups such that they preserve *k*-coverage. To this purpose, for each mobility step, sensors define randomly groups of *k* members for each which are not required to be the nearest neighbors. Each group has a leader which defines mobility steps. The remaining members of the group take into account this choice to determine, in turn, their next mobility step. By this manner, each sensor's mobility step depends on his integrating group. Further, a sensor may move from one group to another in each mobility step. This model enables the definition of overlapping *k*-Voronoi groups which increases the guarantee to have a *k*-coverage. Thus, in the aim to guarantee *k*-coverage all along the estimated target path, the following model is defined:

- A group leader is elected from the set of the nearest nodes to the estimated target path and after receiving a mobility instruction. The group leader follows the mobility instruction sent by IS. Otherwise, groups will move away from the target path.
- Each group leader is in charge of gathering group members. It searches increasingly in its neighborhood.
- A member chooses to belong to a group as long as it does not receive a mobility instruction from an IS. Otherwise, a mobility instruction is prioritized.
- For a mobility step, a member could only belong to a single group. It may then move to another group for further mobility steps.
- A node may act as a group leader as long as it receives mobility instructions from the IS.

5 EXTENSION TO MULTI-TARGET TRACKING

The advanced mobility model have defined the probable zones of a target presence and drives sensors towards these zones. Thus, a mobility instruction is clearly defined and weighted according to the probability of the target presence. In the case of multiple targets, a sensor may receive different mobility instructions and then it chooses which to follow. Nevertheless, this may lead to uneven sensors distributions. Hence, some targets may be not sufficiently covered especially when the number of sensors is not enough to cover all the targets' estimated locations. For these reasons, we propose in the following two techniques enabling the extension of the advanced mobility model for multi-target tracking.

5.1 Extended Advanced Mobility Model

To guarantee the coverage of multiple targets, we present the modifications introduced to the advanced mobility model. In the presented mobility model, sensors are free to define their next movement, two main situations may be defined. In the first one, sensors follow the mobility instruction driving to the target; so, they move towards the probable zones of target presence. In the second, a sensor chooses another different direction taking him away from these zones. The main idea introduced for multi-target tracking is that even when sensors are in the second situation, they remain in nearby locations increasing the probability to return to the target direction in next mobility steps. This can be fulfilled through the customization of velocity according to the chosen direction in the sense that sensor's mobility velocity increases when sensors moves towards target location and inversely. For this reason, we define two velocity ranges. The first, denoted by $\{V_{hi}\}$, contains the high velocity values while the second, denoted by $\{V_{li}\}$, contains the low velocity values.

The extended advanced mobility model links the sensor velocity to the chosen direction. Thus, mobility probability is defined as follows.

- The probability that a node chooses a given velocity is equal to the probability to choose target direction.
- When receiving mobility instructions, the direction of the nearest targets have the higher probability. Consequently, they are assigned the higher probability velocity values.
- Three subsets of velocity values may be defined: (1) a velocity value V_t enabling the sensor to reach

the target position in the next mobility step; (2) a velocity value from $\{V_{hi}\}$ when choosing the target direction but not sufficient to reach the target; (3) a velocity value from $\{V_{li}\}$ when choosing an other direction.

The underlying probability distribution is defined as follows:

$$Pr_V(v) = \begin{cases} P((r,\alpha) = (r_t,\alpha_t)) & v = V_t \\ \frac{1}{V_{hi}} P((r,\alpha) = (r_i,\alpha_i)) & v \in \{V_{hi}\} \\ \frac{1}{V_{li}} (1 - P((r,\alpha) = (r_i,\alpha_i)) & v \in \{V_{li}\} \end{cases}$$

5.2 Extended Mobility Model for Multi-target Tracking

In the following, we divide the monitored zones into regions related to the present targets. Let T be the number of the tracked targets. In the following algorithm, we identify the nearest sensors to each target. Then, we move sensors such that target path remains k-covered.

Algorithm 2: Extended group mobility model for multi-target tracking.

```
While (number of sensors in target's Delaunay triangle ¿ threshold)
do
Compute $T$-Voronoi where $T$ is the number of targets
Define the $k$-Voronoi in each $T$-cell
Apply the group mobility model in each cell
```

6 SENSOR ACTIVITY SCHEDULING

In this section, we highlight the potential given by Voronoi diagrams in implementing activity scheduling strategies. We mainly show how sensors that do not contribute effectively in enhancing the coverage degree within a given zone can be detected and therefore turned-off for a laps of time.

Our idea is to exploit the properties of the Voronoi tessellation to implement a distributed algorithm to identify sensor nodes which sensing coverage is already covered by their neighbors. We first give the definition of a redundant sensor.

Definition 6.1. *A sensor $s_i \in S$ is said to be redundant if, and only if:*

$$\emptyset \neq \Gamma(s_i, R_{s_i}) \bigcap \left(\bigcup_{s_j \in \Sigma(s_i)} \Gamma(s_j, R_{s_j}) \right) = \Gamma(s_i, R_{s_i}), \tag{8}$$

where $\Sigma(s_i) = \{s \in S :\ \Gamma(s, R_s) \cap \Gamma(s_i, R_s) \neq \emptyset \wedge s \neq s_i\}$.

The interest, from the energy consumption optimization point of view, of identifying redundant sensors is obvious since such nodes can be turned-off. In fact, the information provided by a redundant sensor about the presence of a hostile target can be obtained from its neighbors. In the rest of the section, we look for a characterization of redundant sensor nodes.

The following proposition gives a necessary and sufficient condition for node redundancy characterization.

Proposition 6.2. *Let S be a set of sensor node positions in $\mathbb{R}^2$ and $X(s_i)$ the set of intersection points corresponding to $s_i \in S$. If s_i is redundant if, and only if:*

$$X(s_i, V(S)) \subset \bigcup_{s_j \in N(s_i, V(S))} \Gamma(s_j, R_{s_j}). \tag{9}$$

Proof. (i) Proof of $\Rightarrow$: If a sensor s_i is redundant, it comes from Proposition 4.1 that:

$$N(s_i, V(S)) \subseteq \Sigma(s_i). \tag{10}$$

Therefore, it can be written that:

$$\bigcup_{s_j \in N(s_i, V(S))} \Gamma(s_j, R_{s_j}) \subseteq \bigcup_{s_j \in \Sigma(s_i)} \Gamma(s_j, R_{s_j}). \tag{11}$$

From Equations 8 and 11, it comes that if s_i is redundant then

$$\Gamma(s_i, R_{s_i}) \bigcap \left(\bigcup_{s_j \in N(s_i, V(S))} \Gamma(s_j, R_{s_j}) \right) = \Gamma(s_i, R_{s_i}).$$

Moreover, Equation 9 gives that $X(s_i, V(S)) \subset \Gamma(s_i, R_{s_i})$. By transitivity of the inclusion operator, we obtain:

$$X(s_i, V(S)) \subset \bigcup_{s_j \in N(s_i, V(S))} \Gamma(s_j, R_{s_j}).$$

(ii) Proof of $\Leftarrow$: Trivial. □

According to the proposition above, if there exists $x_j \in X(s_i)$ such that $x_j \notin \Gamma(s_i, R_{s_i})$, then s_i is not redundant. Consequently, we propose an algorithm for stating whether a node is redundant or not.

The following corollary extend the result of Proposition 6.2 to the case where a k-coverage of the monitored zone is needed. Obviously, the definition of redundancy should be slightly modified in this case to encompass sensor nodes whose sensing coverage is totally k-covered.

Corollary 6.3. *Let s_i in S be a sensor node. For every x_j in $X(s_i, V(S))$, if $|\{x_j\} \bigcap \left(\bigcup_{s_k \in N(s_i, V(S \setminus \{s_i\})} \Gamma(s_k, R_{s_i}) \right) | < k$, then s_i is not redundant.*

Algorithm 3: Redundant sensor discovery.

```
$\forall s_i \in S$
{ Compute $N(s_i, V(S))$;
Generate $V(N(s_i, V(S)) \setminus \{s_i\}))$;
Compute $X(s_i, V(N(s_i, V(S)) \setminus \{s_i\})))$;
$\forall x_j \in X(s_i, V(N(s_i, V(S)) \setminus \{s_i\})))$
{ $r$:=0;
$\forall s_k \in N(s_i, V(S))$
{ if $(x_j \notin \Gamma(s_k, R_{s_i}))$ then
$r$:=1; } }
if ($r$=1) then
$s_i$ is not redundant;
else
$s_i$ is redundant; }
```

Using this corollary, the strategy defined in the above algorithm remains effective in sensitive contexts where the monitored area should be k-covered.

7 K-SECURITY IN WIRELESS SENSOR NETWORKS

In this section, we will build on the security model presented in (Sliti et al., 2008) to inroduce an optimized $k-$security model. For this purpose, we use higher-order Voronoi diagrams to develop a k out of n threshold signature scheme. It allows any subset of k sensor nodes defined in a voronoi cell to generate a valid signature. Conversely, any subset of k' individual signatures does not constitute a valid signature if $k' < k$. In fact, since the monitored zone is subdivided into Voronoi tesselation, each $k-$Voronoi cell defines the nearest region to its k generators which means that an eventual event occuring in this region should be detected by the k sensors.

In the following, we extend the well-known Shoup threshold cryptosystem (Shoup, 2000) to the elliptic curve context.

Let F_p be a prime finite field so that p is a prime number and let $a, b \in F_p$ satisfying $4a^3 + 27b^2 \neq 0$. An elliptic curve $E_p(a,b)$ over F_p is defined by the set of solutions of the following equation (called Weierstrass equation):

$$y^2 \equiv x^3 + ax + b\,(mod\, p), \tag{12}$$

together with an extra point O called the point at infinity.

Cryptographic schemes based on ECC rely on scalar multiplication of elliptic curve points. Given an integer r and a point $P \in E_p(a,b)$, scalar multiplication is the process of adding P to itself r times. The result of this scalar multiplication is denoted $r.P$.

7.1 Distributed Key Management Scheme

To implement the prposed $k-$security model, the sensor network should be nriched by a Public Key Infrastructure (PKI) that will be in charge of managing the keys ans the certificates used in the various signature management phases. The basic PKI parameters, including the number of CAs, the trust relationships between these CAs, as well as the certificate lifetime will obviously vary according to the features of the monitored battlefield and the nature of the military mission. As far as we are concerned, we illustrate the different cryptographic functionnalities in the simple case where only one CA is considered. In fact, these functionnalities should not change when being applied in a generic context where multiple CAs may be considered.

Key Generation

The key generation phase is performed by the CA as follows:

- generate randomly two large primes $p, q \in \mathbb{P}$, where $\mathbb{P}$ is the set of prime integers, and $p = 2p' + 1$, $q = 2q' + 1$, with p' and q' are themselves primes. Denote $t = pq$ and $m = p'q'$.

- generate the public exponent $e \in \mathbb{P}$ such that $e > n$ (n is the number of private keys). The public key is $\pi = (t, e)$.

- compute $d \in Z$ such that de $de \equiv 1 \bmod m$.

- build the polynomial $f(X) = \sum_{i=0}^{k-1} a_i X^i \in Z[X]$ such that $f(0) = d$.

- compute, for $1 \leq i \leq n$, $\kappa_i = f(i) \bmod m$. The integer κ_i is the private key of sensor node s_i.

Key Revocation

Key revocation may occur in two cases:

1. The sensor node is physically accessed by unauthorized party. In this case, th sensor node should ask for the revocation of its certificate before triggering the tamper-proof functionnalities. To this end, a secret parameter that should have been preloaded by the CA before the deployment of the WSN should have been sent to the CA in oreder to revoke the corresponding certificate. Once this task has been successfully conducted, the critical components of the sensor node are intentionally self-destructed.

2. The sensor node is suspected to be compromised. The suspected node should first be probed to state whether it consists in an intruder node that has spoofed the identity of the legitimate node or it

is a sensor node that has fallen under the control of non-authorized parties. To this purpose, challenge-response authentication messages and radio fingerprinting can be used. If the suspected node is a compromised one, a tolerance state, where the sensor node can forward packets without issuing alert messages, is initiated. During this state, the behavior of the potentially compromized node is monitored to detect the anomalies that it may generate. At the end of the tolerance interval, the suspected node is either rehabilited or irreversibly discarded. From the certificate management point of view, this pre-supposes that during the tolerance state, the certificate of the suspected node is suspended (i.e. subjected to a temporary revocation). In addition, when the node is irreversibly de-activated, the certificate is definitively revoked.

Key Renewal

It may occur that some sensor nodes are recuperated and used for future surveillance missions. In such cases, the key credentials are removed from the sensor nodes and once the old certificate has been checked for revocation, the CA generates a new private key for the node of interest and uploads the corresponding certificate in a physically protected zone of the storage memory.

Key Distribution

Unlike traditional PKI frameworks, only one key distribution scheme is considered in our context. It consists in the secure physical upload of the cryptographic credentials by the CA. To implement this scheme a secure procedure letting the CA accessing the sensitive zones of the non-volatile memory of the sensor node should be available. This procedure should be made available by the manufacturer and the underlying access credentials should be submitted directly to the CA.

7.2 Intermediate Signature Generation and Verification Phase

An event V that may be related to the detection of a hostile target must be converted to a point $P = (v_1, v_2)$ in $E_m(a,b)$. The conversion of a message to an ECC point is done according to the approach proposed in (Hankerson et al., 2004). Then, a sensor node s_i can generate its individual signature (σ_i^x, σ_i^y) using its private key according to the following equation.

$$(\sigma_i^x, \sigma_i^y) = 2n!\kappa_i(v_1, v_2) \bmod m.$$

This signature will then be forwarded to the corresponding core sensor through a set of relay nodes (belonging to the sensing layer) denoted by $R = \{r_1, .., r_u\} \subseteq \{s_1, .., s_n\}$. An important functionality that could be implemented by the relaying sensors is to identify and withdraw false alert messages. This feature is called intermediate verification. We suppose that a secret integer γ is broadcasted by the core node (this broadcast should be enciphered using a threshold encryption algorithm). Hence, node s_i computes the point $\tilde{P} = \gamma.(\sigma_i^x, \sigma_i^y) \bmod t$. Instead of sending only the signed message to the core sensor, s_i also sends the integer ω, which stands for the order of $\tilde{P}$ (i.e., $\omega.\tilde{P} = O$).

To check the validity of an individual signature, node $r_j \in R$ calculates the point Q as follows.

$$Q = \omega\gamma.(\sigma_i^x, \sigma_i^y) \bmod t. \qquad (13)$$

The verification succeeds if, and only if, $Q = O$.

Clearly, the robustness of the intermediate verification scheme builds upon the complexity, for the intruder, to determine γ. Effectively, computing γ knowing (σ_i^x, σ_i^y) and ω involves discrete logarithm computation within $E_t(a,b)$. This problem has been shown to be intractable in (SEC 1, 2000).

7.3 *k*-Vornoi Cell Signature Verification

Supposing that a core sensor has collected k signatures generated by a subset of sensor nodes $(s_i)_{i \in S}$, where $S = \{i_1, .., i_k\}$, the objective is to verify whether all these signatures are valid. To this purpose, we compute:

$$w = \Sigma_{i=1}^{k} 2.\lambda_{0,i}.(\sigma_i^x, \sigma_i^y) \bmod t,$$

where

$$\lambda_{i,j}^S = n! \frac{\Pi_{j' \in S}(i - j')}{\Pi_{j' \in S}(j - j')}.$$

To verify the validity of the global signature, the core sensor checks if:

$$ew = 4n!^2 (v_1, v_2) \bmod t \qquad (14)$$

A proof of correctness is given in the following. We first develop the expression of w.

$$w = 4n! \Sigma_{i=1}^{k} \lambda_{0,i_j} (f(i) \bmod m)(v_1, v_2) \bmod t$$

From the properties of the determinant of the Vandermonde matrix, we obtain:

$$n! f(i) = \Sigma_{j \in S} \lambda_{i,j}^S f(j) \bmod m.$$

Therefore, it can be written that:

$$w = 4n!^2 f(0)(v_1, v_2) \bmod t = 4n!^2 d (v_1, v_2) \bmod t$$

$$ew = 4n!^2 (v_1, v_2) \bmod t.$$

8 CONCLUSIONS AND POTENTIAL EXTENSIONS

This paper presented two Voronoi-based mobility models for target tracking using WSNs. The key advantage of these models is that they encompass the potential target position in the construction of the mobility instructions. This ensures that the locations where the target is most probable to be are more covered than the rest of the monitored area. We also proposed a redundancy discovery technique to enhance the WSN cost-effectiveness. An enhancement of this work to build a multi-target tracking framework is currently under development.

REFERENCES

C. Gui and P. Mohapatra, ”Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks,” Proc. ACM MobiCom ’04, pp. 129-143, Sept. 2004.

M. Hamdi, N. Boudriga, M. S. Obaidat, *WHOMoVeS: An optimized broadband sensor network for military vehicle tracking*, International Journal of Communication Systems, Vol. 21 , Issue 3, pp. 277-300, ISSN:1074-5351, 2008.

L. Wang, H. Shen, Z. Chen, and Y. Lin, *Voronoi Tessellation Based Rapid Coverage Decision Algorithm for Wireless Sensor Networks*, Lecture Notes in Computer Science, Ubiquitous Intelligence and Computing, Springer, 2007.

I. Stojmenovic, A. K. Ruhil, D.K. Lobiyal, *Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks*, Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC03), KEMER - ANTALYA, Turkey, 2003.

Marcos Augusto M. Vieira, Luiz Filipe M. Vieira, Linnyer B. Ruiz, Antonio A. F. Loureiro, Antonio O. Fernandes, Nogueira Nogueira, *Scheduling Nodes in Wireless Sensor Networks: A Voronoi Approach*, lcn, p. 423, 28th Annual IEEE International Conference on Local Computer Networks (LCN’03), 2003.

W. Wang, V. Srinivasan, and K-C. Chua, *Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks*, MobiCom07, September 914, 2007, Montral, Qubec, Canada.

M. Sliti, M. Hamdi, N. Boudriga. *An Elliptic Threshold Signature Framework for k-Security in Wireless Sensor Networks*, ICECS, 2008.

V. Shoup, *Practical Threshold Signatures Proc*, Int’l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT ’00), 2000.

D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, 2004.

Standards for Efficient Cryptography, SEC 1: *Elliptic Curve Cryptography*, Version 1.0, September 2000.