Sensors Deployment Strategies for Rescue Applications in Wireless Sensor Networks

Inès El Korbi, Nesrine Ben meriem, Leila Azouz Saidane
CRISTAL Laboratory
National School of Computer Science
University of Manouba, Tunisia
e-mails: ines.korbi@ensi.rnu.tn, bmariem.nesrine@gmail.com, Leila.saidane@ensi.rnu.tn

Abstract— **Wireless sensor networks (WSNs) become a major tool for various security and surveillance applications to detect and monitor environmental changes, to control vehicle traffic, etc. For all these capabilities, we propose in this paper to use WSNs in rescue applications. These applications are critical in terms of network response time when disaster strikes. The network response time is the time required by sensor nodes to detect victims' positions and forward them to the sink node. In this paper, we consider two sensor nodes deployment strategies: linear and circular strategies. By deployment strategies, we mean initial nodes positioning and nodes movement procedures. Therefore, we simulate these two strategies under the WSNet simulator and derived the Monitored Area Sweep Time (MAST) (i.e., the time required by sensor nodes to scan all the area coordinates to find victims) and the energy dissipation due to node mobility. Simulation results are verified using analytical expressions of the zone sweep time and the energy dissipation. Finally, we extend the behavior of the sensor nodes deployment strategies to support nodes communication so that mobile sensors can forward victims' positions to the sink node.**

Keywords- wireless sensor network; mobility; sensor nodes deployment strategies; rescue applications.

I. INTRODUCTION

Recent advances in miniaturization such a low power circuit design and low powered wireless communications make it possible the emergence of a new kind of equipments: sensor nodes. The sensor node is a few cubic centimeters device with various capabilities such as simple wireless communication, minimal computation facilities, and sensing of the physical environment. Typical sensing tasks are temperature, light, vibration, sound, radiation, etc. The above characteristics of sensor nodes, allowed the usage of Wireless Sensor Networks (WSNs) [3], [13] in different fields on applications.

Indeed, WSNs were initially designed for military applications, such as battlefield surveillance and enemy tracking. Now, they are used in many industrial and civilian application areas, such as industrial process monitoring, environment and habitat monitoring, healthcare applications and traffic control [10].

In this paper, we focus on area monitoring applications and particularly on rescue applications to detect victims' positions when hazardous events occur (seism, fire, explosion, etc.) [5], [11]. This kind of applications imposes strict time requirements and the efficiency of WSN deployment procedure depends on the network response time. The network response time corresponds to the time required by the sensors deployed in the monitored area to detect victims' positions and notify the sink node accordingly. Therefore, the faster, the victims are localized, the faster rescue personal can perform their tasks and help victims. In our work, we focus on sensor nodes with motion capabilities. Mobile sensors are initially deployed over the monitored area. They remain in their fixed positions until receiving a hazardous event message from the sink node that triggers their movement and the victims' detection procedure.

Therefore, we propose in this paper, two sensor nodes deployment strategies to detect victims in the wireless network. These techniques are called circular and linear strategies. For these two techniques, we propose algorithms to both initially place sensor nodes and then moving them within the monitored area. Then, we simulate both linear and circular strategies using the WSNet simulator [14]. From simulation results, we derive the Monitored Area Sweep Time (MAST) which corresponds the time required by the sensor nodes to scan all the monitored area coordinates to localize victims. Then, we evaluate the energy consumption due to node mobility for both techniques. Simulation results are verified analytically using both sweep time and mobility based energy consumption expressions. Analytical results verify that the nodes initial deployment and movement algorithms are correctly implemented under the WSNet simulator. Hence, these algorithms could be extended to support communication aspects between sensor nodes for victim detection purposes.

Indeed, in the last part of the paper, we focus on victim detection mechanism by introducing a message exchange procedure between the different sensors within the network to localize victims and communicate their positions to the sink node. We then evaluate the energy consumption due to both communication and mobility and compare the MAST value to the last victim detection time (the time elapsed until the last victim in the monitored area is detected) for different sizes of the monitored area.

In the rest of the paper, we review in Section 2 the works related to our study. Section 3 presents the concepts and the algorithms of the sensors nodes deployment strategies. In Section 4, we simulate our strategies using the WSNet network simulator [14] and derive expressions of the monitored zone sweep time. We also derive energy consumption curves due to sensors mobility. Simulation results are verified analytically using expressions of both sweep time and energy consumption due to mobility. Section 5, presents the victim detection mechanism within the monitored area by ensuring communication between the sink node and the mobile sensors. We therefore derive the resulting energy consumption due to both mobility and communication and compare the MAST value to the last victim detection time. We conclude the paper and present our future work in Section 6.

II. Related Works

Area monitoring applications use WSNs either for measuring and surveying purposes or for reporting various types of activities and events. Therefore, the area coverage criterion has to be met. A point is covered by a sensor if it is within its sensing range. In [1], authors design a distributed self deployment algorithm for coverage calculations in mobile sensor networks and consider various performance metrics, like coverage and uniformity. The work in [15] assumes that a cluster head is available to collect information and determine the target location of the mobile sensors. Sensor deployment has also been addressed in the field of robotics [8], [9], where sensors are deployed one by one, utilizing the location information of previously deployed sensors.

After being initially deployed over the monitored area, sensor nodes have to sweep the monitored zone to detect victims. Sweeping algorithms have received much attention in the past few years. The performance of these algorithms can be evaluated from various aspects, including the achieved coverage percent, the number of deployed sensors and the time required for the sweeping. In [2], authors investigate the problem of how to optimally move mobile sensors lying within a region to the perimeter of that region to detect intruders. In [6], Rekleitis et al. propose an approach to sweep all the destination zones. They assume that mobile nodes can communicate with each other and know their positions. The area is divided into stripes, with each mobile node taking care of one stripe. Wong et al. [12] use topological mapping to sweep the destination area. They make cell decomposition and cover each cell by a zigzag pattern. Batalin and Sukhatme [7] propose a decentralized method and present the frequency coverage metric to evaluate the quality of sweep coverage. The common challenges of these studies are the coverage ratio and energy consumption. Therefore, this paper aims to develop two efficient sweeping algorithms (the linear and circular sweeping algorithms) such that the number of deployed sensor is as few as possible while the monitored region is guaranteed to have full coverage.

In the next Section, we introduce two sensor nodes deployment techniques where sensors can move autonomously to sweep the monitored area.

III. Sensor Nodes Delployment Strategies

In this Section, we define two sensors deployment strategies to localize victims in rescue applications. The deployment strategies define at the same time the initial nodes locations and nodes movement when a hazardous event occurs. These two components of the sensors deployment strategies will definitely determine the efficiency of a strategy and its response time to perform victim localization. We first define concepts of linear and circular strategies.

A. Fundamental concepts of linear and circular strategies

When deploying sensor nodes within the monitored area, two basic criteria have to be met:

- At any time, communication between every sensor node and the sink node has to be maintained.
- All the points of the monitored area have to be covered by the wireless sensors.

Therefore, we define the following quantities:

- R_S : The node sensing range
- R_C : The node communication range, $R_C = 2R_S$
- N : The number of mobile sensors in the monitored area.
- p : The sink node placed at the center of the monitored zone and communicates victims' locations to the centralized system. The sink node's coordinates are initially know and given by (x_p, y_p). Fixing the sink position will simplify the communication mechanisms between the sink and the mobile sensor nodes.
- D : The diagonal of the zone to be monitored by the sensors. In the rest of the study, we consider that the monitored zone is square shaped. This assumption can be easily extended to a rectangular zone with height x and width y, where $D = y\sqrt{2}$ (and $y = \max(x, y)$).

1) The linear strategy

The linear strategy illustrated in Figure 1 consists in placing the wireless sensor nodes on axes horizontally and vertically.

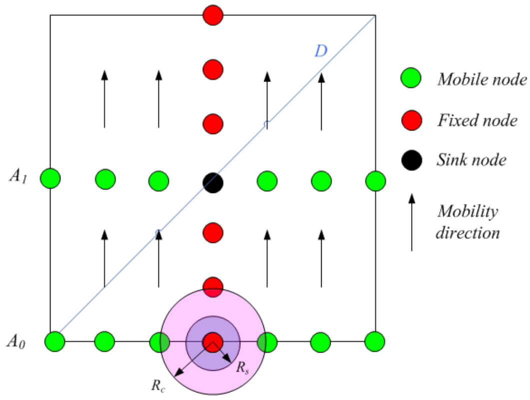

Figure 1. Sensor nodes initial positions according to the linear strategy

Therefore, we define two sensor categories:

- *Fixed sensors*: are placed vertically on the monitored area aligned with the sink node on its both sides to keep communication with the mobile nodes when they are moving.

- *Mobile sensors*: are placed on horizontal axes such as the distance that separates two adjacent sensors on a horizontal axis is always R_c . When an axis is saturated, we place another horizontal axis parallel to the previous axis.

The number of horizontal axes depends on the number of sensors to be placed on the monitored area.

Horizontal axes are numbered from A_0 to A_{K-1} , where K is the number of mobile axes. Axis A_0 sensors are situated on the lowest boundary side of the monitored area (Figure 1). When an event happens, the mobile nodes move along vertical axes in bottom up direction until sensors on a given horizontal axis A_i reach the position of their successors' nodes on the next axis A_{i+1} . We say that $n_{i+1,j}$ is $n_{i,j}$ successor's node if $n_{i,j}$ is placed on axis A_i and $n_{i+1,j}$ is placed on axis A_{i+1} , such as $n_{i,j}$ and $n_{i+1,j}$ have the same x_j abscissa. Nodes stop moving when the nodes on axis A_{K-1} reach the upper boundary side of the monitored area.

When moving along vertical axes, a node situated on a given horizontal axis will forward the information of a victim location to its neighbor on the same horizontal axis until its message reaches one of the fixed sensors that will forward the received message vertically till it reaches the sink node. As in [11], we assume that the sensors are location aware (i.e., they know their geometric coordinates when they are initially deployed). When moving, the new positions are determined as a function of the old ones.

2) The circular strategy

The circular strategy consists to initially place the N mobile sensors around the sink node on axes as in Figure 2. The distance separating two adjacent nodes on the same axis is equal to R_c . Each node on a given axis defines a level.

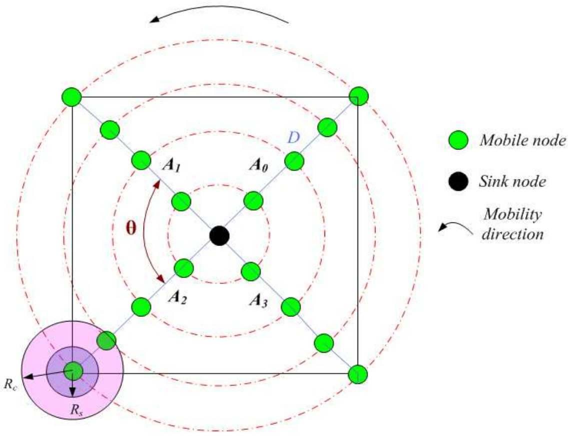

Figure 2. Sensor nodes initial positions according to the circular strategy

A sensor node is said to be on level i , if the distance that separates this node from the sink is equal to $R_c \times i$.

Two adjacent nodes on a level i are separated by a distance equal to $R_c \times \theta$, where θ is the angle between two adjacent axes. θ is equal to $2\pi / K$, where K is the number of axes within the network.

When a hazardous event occurs within the sensor network, each sensor node turns around the sink in a counter-clockwise way. Sensors on the same level i turn around the sink simultaneously at the same speed. Moreover, nodes on the same axis (belonging to different levels) have to progress together to keep communication with the sink node. Hence, when moving, nodes have to be arranged along their original axes at any point of time. To keep axes structure, two nodes $n_{i,l}$ and $n_{j,l}$ belonging to the same axis A_l and placed on levels i and j have to perform the same angular distance θ_r , between two time instants 0 and t_r where θ_r is given by:

$$\theta_r = \frac{D_i}{R_i} = \frac{D_j}{R_j} \qquad (1)$$

where D_i (respectively D_j) is the distance (in meters) traversed by a sensor on level i (respectively on level j) and R_i (respectively R_j) is the distance separating the node $n_{i,l}$ (respectively $n_{j,l}$) from the sink. Hence, according to equation (1), D_j will be equal to:

$$D_j = \frac{R_j}{R_i} D_i \qquad (2)$$

As the two nodes $n_{i,l}$ and $n_{j,l}$ have to go through distances D_i and D_j within the same time interval $[0, t_r]$, the velocity of node $n_{j,l}$ have to be $\frac{R_j}{R_i}$ times the one of node $n_{i,l}$. Finally, sensor nodes stop moving, when each sensor performed an angular distance equal to θ .

If a victim was detected while moving around the sink node, a mobile node $n_{i,l}$ located on axis A_l and level i , will forward the victim's coordinates to the node $n_{i-1,l}$ located on the same axis A_l and the lower level $i-1$. This procedure is repeated until the victim coordinates reach the sink node.

B. Proposed algorithms for linear and circular strategies

1) The linear strategy algorithms

In the following, we detail the algorithms corresponding to the initial nodes positioning and nodes movement algorithms using the linear strategy. Algorithms related to nodes communication to detect victims' positions and forward them to the sink node are detailed in Section 5. We define:

- N_f : The number of fixed sensors. It also corresponds to the number of sensors per mobile axis.

- K: The number of horizontal axes.
- V : The velocity of a mobile node.

Algorithm 1: Initial Sensor Nodes Positioning

Input: N, D, V, R_c and (x_p, y_p)

Output: *NList*: list of mobile and fixed sensor nodes

```
N_f ←ceil( D/(2*sqrt(2)*R_c) )*2;
K←floor( (N − N_f)/ N_f )
Sn←new Node(); Sn →x ← x_p ; Sn →y ← y_p
Add(Sn, NList);
h1← -1; h2← -1
//Deploying fixed nodes
For i from 1 to N_f do
    Sn ←new Node(); h2← h2*h1
    Sn →x ← x_p ; Sn →y ← h2* Rc*i
    Sn →speed ← V ; Add(Sn, NList);
End
//Deploying mobile nodes
For i from 1 to K do
  For j from 1 to N_f do
    Sn ←new Node();
    h2← h2*h1
    Sn →x ← x_p + h2*j* R_c
    Sn →y ← D/( sqrt(2) *i)
    Sn→speed ← V ; Add(Sn, NList);
  End; End
```

Since we consider floor and ceil functions, only $N_f \times (K+1)$ sensor nodes will be deployed on fixed and mobile axes and participate in the victim detection procedure. The remaining $(N - N_f \times (K+1))$ sensors will be placed randomly and wouldn't be used. When a hazardous event occurs, nodes move along vertical axes for a distance $Dis = D/(K\sqrt{2})$. The following algorithm illustrates mobile nodes movement:

Algorithm 2: Sensor Nodes Movement

Input: *N*; *D*; R_c and *NList*: the list of sensor nodes

TS: the node movement time step in seconds.

Output: Monitored area sweeped

```
N_f ←ceil( D/(2*sqrt(2)*R_c) )*2;
K←floor( (N − N_f)/N_f )
Dis←D/(K*sqrt(2) ); DSn← 0
Sn← First(NList);
For i from 1 to N_f do Sn ←Next( NList) End
While (DSn < Dis) do
    While ( Sn) do
        Sn→move(Sn→x , Sn→y + V* TS, TS)
        Sn ←Next( NList)
    End
    DSn ← DSn + V* TS;
End
```

In the above algorithm, each sensor calls its own function *move(a, b, ts)* to go from its current position to the *(a, b)* position within *ts* time interval. The *move* function also updates node's position when the *(a, b)* are reached.

2) The circular strategy algorithms

Before detailing initial positions placement and movement algorithms of the circular strategy, we define:

- N_a : The number of sensors per axis.
- K : The number of axes.
- θ: The rotation angle defining the angular distance to be performed by each sensor node.
- V : The velocity of a sensor node at level 1 (the nearest level to the sink node). According to equation (2), nodes belonging to level i progress at a speed $V_i = V \times i$ since level i nodes are separated from sink node by a distance equal to $R_c \times i$.

Algorithm 3: Initial Sensor Nodes Positioning

Input: *N*; *D*; V ; R_c and (x_p, y_p)

Output: NList: Sensor nodes list

```
N_a ←ceil(D/(2* R_c )); K←floor(N/ N_a )
θ ← 2π /K; Sn→x ← x_p ; Sn→y ← y_p
Add(Sn, NList);
For i from 1 to K do
   For j from 1 to N_a do
      Sn ←new_Node();
      Sn→x ←sin((i-1)* θ )* R_c *j+ x_p
      Sn→y ←cos((i-1)* θ )* R_c *j+ y_p
      Sn→speed ← V *j
      Add(Sn, NList);
   End; End
```

When a hazardous event occurs, nodes move around the sink node until they go through an angular distance of θ. The following algorithm illustrates the sensor nodes movement without considering communication aspect between the nodes which will be detailed in Section 5. We use the *move* function introduced in algorithm 2.

Algorithm 4: Sensor Nodes Movement

Input: *NList*; *N*; R_c ; *D*; V and (x_p, y_p)

TS: the node movement time step in seconds.

Output: Monitored area sweeped

 ISBN:978-1-61208-113-7

$N_a \leftarrow$floor(D/(2* R_c)); $K \leftarrow$ceil(N/ N_a)
$\theta \leftarrow 2\pi /K$; $\theta_f \leftarrow 0$; $\theta_1 \leftarrow V * TS/ R_c$
While ($\theta_f < \theta$) **do**
Sn← First(*NList*); *Sn*← Next (*NList*)
$\theta_f \leftarrow \theta_f + \theta_1$
  For *i* **from** 1 **to** K **do**
    For *j* **from** 1 **to** N_a **do**
      $a \leftarrow \sin((i\text{-}1)^* \theta + \theta f)^* R_c {}^*j + x_p$
      $b \leftarrow \cos((i\text{-}1)^* \theta + \theta f)^* R_c {}^*j + y_p$
      Sn→move(a , b, TS)
      Sn← Next (*NList*)
    End; **End**; **End**

IV. PEFORMANCE EVALUATION OF LIENAR AND CIRCULAR STRATEGIES

In this Section, we propose to evaluate the performance of the linear and circular strategies. Indeed, we evaluate the Monitored Area Sweep Time (MAST) which corresponds to the time required by the sensor nodes to scan all the monitored area coordinates to locate victims. Moreover, as in the circular strategy sensor nodes progress at different speeds on the different levels of a given axis, we also propose to evaluate the energy consumption of linear and circular strategies due to mobility. To evaluate the sensor nodes deployment strategies, we implement the algorithms 1 to 4 defined in Section 2 under the WSNet [3] simulator, an event driven simulator dedicated to wireless sensor networks. WSNet defines models written in C for the different network layers Simulation scripts can be configured through xml files. Using the WSNet-replay tool, we present in Figure 3 the initial node deployment screen using the circular strategy (The number of sensors N=40).

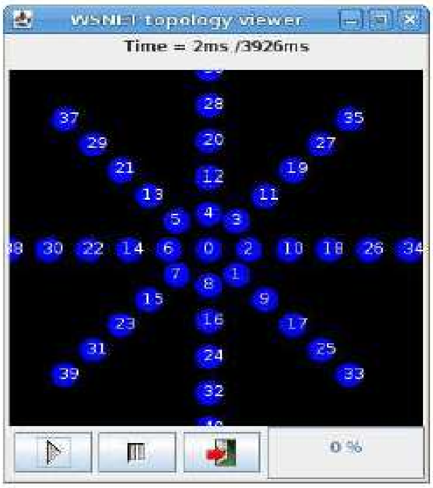

Figure 3. WSnet-replay screen for the circular strategy

In the next Section, we focus on area sweep time and energy consumption performance evaluation.

A. The Monitored Area Sweep Time

To derive results of the MAST parameter, we consider the communication range R_c equal to 30m, the velocity of mobile nodes in the linear strategy and level 1 nodes in the circular strategy equal to 2 m/s. In Figure 2, we fix the number N of sensor nodes deployed on the monitored area to 50 sensors, and depict the monitored zone sweep time as a function of the diagonal D of the monitored zone for both linear and circular strategies.

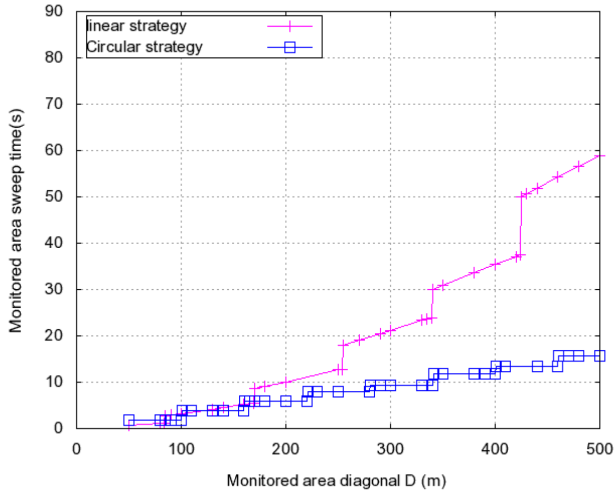

Figure 4. Monitoric Area Sweep Time

The curves show that with linear and circular strategies, we obtain step based sweep time values. This behavior can be explained by the use of *ceil()* and *floor()* functions. Moreover, curves show that the circular strategy sweep time growth is smoother than the one of the linear strategy as D increases. We can therefore conclude that it would be better to use the circular strategy when large areas are monitored. However, we can't conclude which strategy is better without considering the energy dissipation effect caused by nodes mobility.

B. Energy consumption due to mobility

In this paragraph, we focus on energy consumption caused by nodes mobility. Indeed, a pretty simple model of energy consumption due to mobility was introduced in [4] and states that the energy drops linearly with the distance (27.96 J/m).

To implement this energy dissipation model in simulation scenarios, we consider that each sensor node has an initial energy of 10^6 joules and we trigger a periodic event that computes the distance between the old node position and the current node position and evaluates the energy dissipation value. Figure 5 depicts the energy consumption of both linear and circular strategies as a function of D. Curves show that the energy consumed by the linear strategy is smaller than the one consumed by the circular strategy since nodes in circular

ISBN:978-1-61208-113-7

strategy progress at different speeds. We can therefore conclude from sweep time and energy consumption values, that the circular sensors deployment strategy offers a better zone sweep time than the linear strategy at the expanse of high energy consumption.

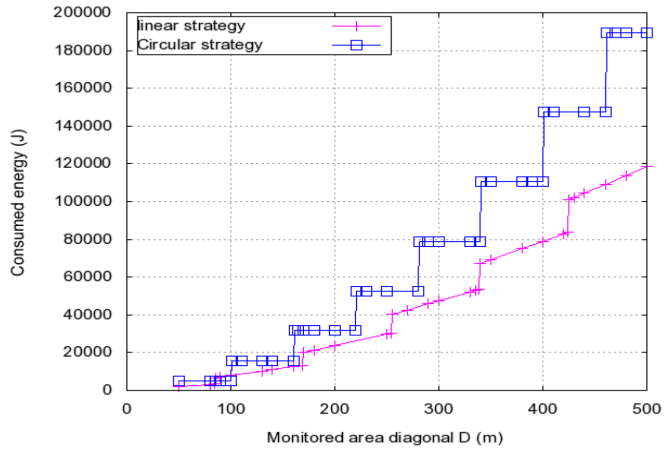

Figure 5. Energy consumption caused by sensors mobility

C. Verification of simulation results

In the above paragraphs, we depicted MAST and energy consumption values obtained by simulation. In this paragraph, we propose to verify the simulation results by considering analytical expressions of the monitored area sweep time and energy consumption due to mobility as a function of the number of sensors *N*, the monitored area diagonal *D*, the communication range R_c and the velocity *V*.

Indeed, in the linear strategy, as sensor nodes on horizontal axes progress simultaneously, the time required by sensor nodes to sweep all the monitored area is the time required by each mobile sensor on a horizontal axis to go through a $D/\left(K\sqrt{2}\right)$ distance. Therefore, the MAST parameter analytical expression is given by:

$$MAST = \frac{D}{\sqrt{2} * V * \left\lfloor N/2\left\lceil D/2\sqrt{2}/R_c \right\rceil - 1\right\rfloor} \quad (3)$$

In the circular strategy, as nodes located on the same axis and on different levels have to progress simultaneously, the monitored area sweep time is defined by the time required by a node on level 1 to go through an angular distance of $\frac{2\pi}{K}$ with the velocity V, where K is the number of axes. Therefore:

$$MAST = \frac{2\pi R_c}{\left\lfloor N/\left\lceil D/2R_c \right\rceil \right\rfloor V} \quad (4)$$

where $\left\lfloor N/\left\lceil D/2R_c \right\rceil \right\rfloor$ is the analytical expression of the number of axes K. Using the same parameters values as for simulation scenarios, we depict in Figure 6 analytical and simulation results of the monitored zone sweep time as a function of the diagonal *D*. Curves show that for both strategies, analytical and simulation results coincide. In the same way, we can derive energy dissipation analytical expressions caused by nodes mobility.

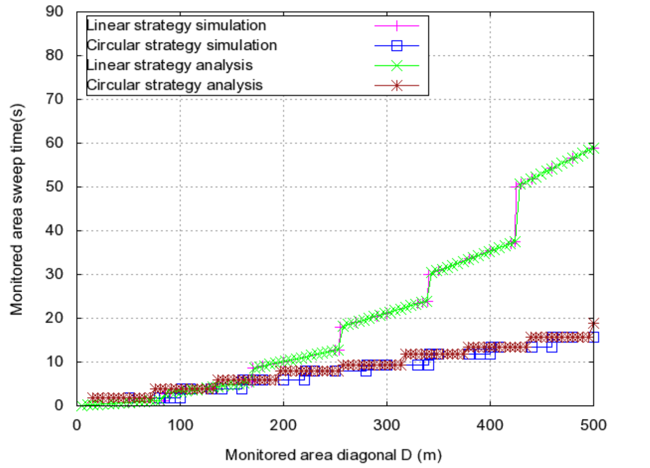

Figure 6. Analytical and simulation results of the monitored area sweep time

Indeed, in the absence of node communication, the energy decreases linearly with the distance (α joules/m). If the linear strategy is used, all the $(K * N_f)$ mobile sensors, go through a $D/\left(K\sqrt{2}\right)$ distance. Therefore, the whole network consumption energy (WNCE) is:

$$WNCE = \frac{D * \alpha * K * N_f}{\sqrt{2} * K} = \frac{D\alpha}{\sqrt{2}} * 2\left\lceil D/2\sqrt{2}/R_c \right\rceil \quad (5)$$

When the circular strategy is considered, the energy consumed by level *i* sensors is *i* times the one consumed by level 1 sensors. Therefore:

$$\begin{aligned} WNCE &= K\left(R_c\theta + 2R_c\theta + 3R_c\theta + \ldots N_a R_c\theta\right)\alpha \\ &= 2\pi R_c \frac{N_a\left(N_a+1\right)\alpha}{2} = \pi R_c \alpha \left\lceil D/2R_c \right\rceil \left(\left\lceil D/2R_c \right\rceil + 1\right) \end{aligned} \quad (6)$$

In Figure 7, we depict the energy consumption due to mobility for linear and circular strategies using both analysis and simulation results (the dissipation factor α= 27.96 J/m).

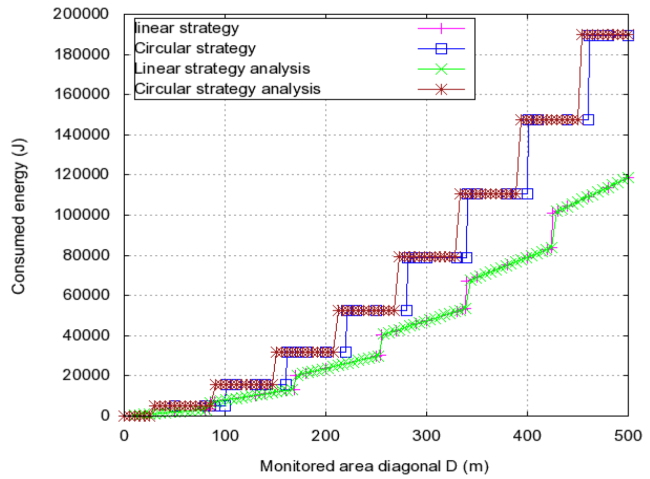

Figure 7. Analytical and simulation results of the energy consumption

 ISBN:978-1-61208-113-7

As for the MAST parameter, analytical and simulations results of energy consumption due du mobility coincide. We can therefore conclude that sensor deployment strategies behaviors are correctly implemented under the WSNet simulator and this behavior can be extended to support nodes communications for victims' detection purposes.

V. VICTIMS DETECTION MECHANSIM

As the simulation modules implementation was verified analytically, we propose in this Section to extend the movement algorithms 2 and 4 detailed in Section 3 to support nodes communication so that mobile nodes can notify the sink node of victims' positions when they are detected.

Indeed, in the linear strategy, we propose a new mobility model that ensures communication between sensor nodes when a victim is detected. If a sensor node detects a victim, it sends a message to its neighbor belonging to the same horizontal axis and the nearest one to the fixed sensors. The receiving node will repeat the same procedure until the message reaches one of the vertical sensors that will forward the message to the sink node (*ID=0*). Therefore, the algorithm 2 related to the nodes movement in Section 3 becomes:

Algorithm 5: Sensor Nodes Victims Detection

```
Input: N; Rc; D, V; NList and (xp, yp)
TS: the node movement time step in seconds.
Output: Victims positions detected.
    Nf ←ceil( D/(2*sqrt(2)*Rc))*2;
    K←floor( (N – Nf)/Nf ); Dis←D/(K* sqrt(2) );
    DSn← 0; Sn← First(NList); Sn← Next(NList)
    For i from 1 to Nf do    //Fixed sensors
       VicList ←Detect_victims(Sn)
       if (VicList) then
            msg←New_Msg( ); Add( msg, VicList)
       End if
       if (Sn→y < yp ) then  sg←1 else sg← -1
       End if
       Tx_victim(Sn→x, Sn→y+ D/Rc/sqrt(2)*sg ,
       msg)  ;  Sn ←Next( NList)
    End
    While (DSn < Dis) do   //Mobile sensors
       While (Sn) do VicList ←Detect_victims(Sn)
          if (VicList) then
            msg←New_Msg( ); Add( msg, VicList)
          End if
          if (Sn→x < xp ) then sg←1 else sg← -1
          End if
          Tx_victim(Sn→x + D/ Rc/sqrt(2)*sg ,
          Sn→y, msg) ;
          Sn→move(Sn→x , Sn→y + V* TS, TS)
          Sn ←Next( NList);
       End
       DSn ← DSn +V*TS
    End
```

In algorithm 5, the function *Detect_victims*(*Sn*) detects victims in *Sn* sensing range and returns their positions list. Then, *Sn* calls the recursive function *Tx_victim()* to transmit victims positions to its direct neighbor *Sn+1* on the same horizontal axis. As *Tx_victim()* is a recursive function, *Sn+1* will repeat the same procedure until the message reaches one of the vertical sensors that forwards the message to the sink node using the same function *Tx_victim()*.

In the circular strategy, we propose a new algorithm to ensure communication between sensor nodes. Indeed, when a mobile sensor detects a victim in its sensing range, it sends a message to the lower level node situated on the same axis. Sensor nodes on lower level axis will forward the message until it reaches the sink node (*ID=0*). Therefore, the algorithm 4 related to the nodes movement in Section 3 becomes:

Algorithm 6: Sensor Nodes Victims Detection

```
Input: N; Rc ; D; V ; NList and (xp, yp)
TS: the node movement time step in seconds.
Output: Victims positions detected.
    Na ←floor(D/2* Rc);  K ←ceil(N/ Na )
    θ ← 2π /K;  θf ← 0 ;  θ1 ← V *TS/ Rc
    While ( θf < θ ) do
      Sn← First(NList); Sn← Next (NList)
      θf ← θf + θ1
      For i from 1 to K do
         For j from 1 to Na do
            VicList ←Detect_victims(Sn)
            if (VicList) then
              msg←New_Msg( ); Add( msg, VicList)
            End if
            a ←sin((i-1)* θ+ θf )* Rc *j + xp
            b ←cos((i-1)* θ+ θf )* Rc *j + yp
            a1←sin((i-1)* θ+ θf - θ1 )* Rc *j + xp
            b1←cos((i-1)* θ+ θf - θ1 )* Rc *j + yp
            Tx_victim((a – xp)* (a1 , b1, msg)
            Sn→move(a , b, TS);  Sn← Next (NList)
         End;
    End; End
```

In the same way, the recursive function *Tx_victim()* allows a sensor node *Sn* to forward victims positions to his direct neighbor on the same axis and the level below. Using the recursive *Tx_victim()* function victims' positions are forwarded to the sink node. In table 1, we compare the energy dissipation caused by mobility to the resulting energy dissipation caused by both mobility and communication:

TABLE I. ENERGY CONSUMPTION VALUES

Diagonal D (in m)	*Linear strategy*		*Circular strategy*	
	No communication	*Communication*	*No communication*	*Communication*
100	7902.62	7903.21	5266.56	5266.98
200	23707.87	23708.49	31599.39	31599.86
300	47415.75	47416.31	78998.48	78998.78
400	79026.25	79026.84	110597.88	110598.27
500	118539.38	118539.98	189596.31	189596.80

Table 1 shows that sensors nodes communication has a little impact on energy consumption and the major energy dissipation is due to mobility.

In Figure 8, we consider two simulation scenarios with different number of victims (20 and 80 victims). For these two scenarios, we depict the last victim detection time as a function of the monitored area diagonal *D*.

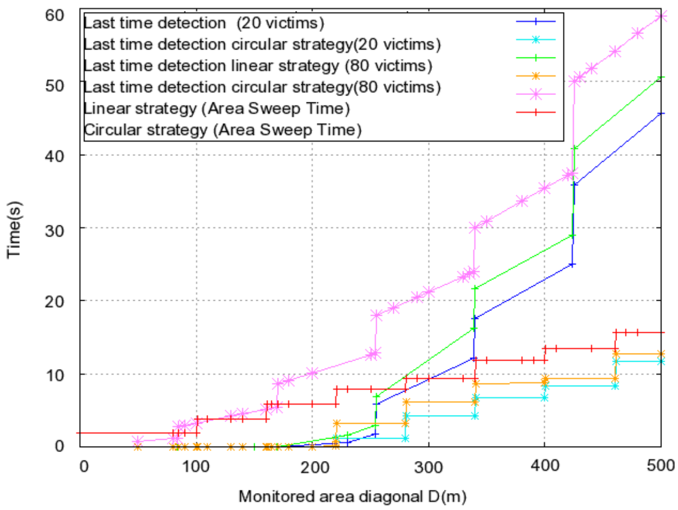

Figure 8. Comparision of last victim detection time and the MAST value

Figure 8 shows that the last victim detection time curves are under the MAST curve for both linear and circular strategies. Moreover, we notice that the last victim detection time curves become closer to the MAST curve as the number of victims increases. Therefore, we conclude that the area sweep time is an upper bound of the time required to detect all the victims within the monitored area and that the last victim detection time approaches the MAST value when the number of victims to be detected increases considerably.

VI. CONCLUSIONS AND FUTURE WORK

A. *Conclusions*

In this paper, we focused on the use of wireless sensor networks in rescue applications. This application consists in deploying a wireless sensor network in a monitored area to locate victims when a disaster occurs. Then, we considered mobile sensor nodes and focused on the way they are initially deployed and how do they move within the monitored area when a hazardous event happens. Therefore, we proposed two sensor deployment strategies: the linear strategy, where sensor nodes are arranged along horizontal and vertical axes and the circular strategy where sensors are arranged in a circular way around the sink node. For these two strategies, we proposed algorithms for initial nodes deployment and nodes movement within the network. Therefore, we implemented the algorithms of both strategies under the WSNet simulator. For These two techniques, we derived the MAST parameter which corresponds to the time required by the sensor network to scan all the monitored area coordinates to find victims. We also derived energy consumption due to mobility as mobile nodes in the circular strategy move at different speed values. Simulations results were easily verified using analytical expressions of the MAST and energy consumption values. The results showed that circular strategy is faster than linear strategy in terms of monitored area sweep time. However, the whole network energy consumption in the circular strategy is greater than the one of the linear strategy. In the last Section, we considered communication between mobile nodes and introduced message exchange mechanisms in both linear and circular strategies. We derived the energy consumption resulting in both nodes mobility and communication. We also verified by simulation that the MAST value is an upper bound of the time required detecting all the victims, which corresponds to the last victim detection time. Moreover, we noticed that the last victim detection time encloses the MAST value as the number of victims increases within the monitored area.

B. *Future work*

In our future work, we propose to evaluate the sensor nodes deployment strategies under more realistic conditions. Indeed, we'll consider that we are in presence of obstacles within the monitored area and we'll propose different solutions so that obstacles could be by bypassed by the sensor nodes. Moreover, when sensor nodes move within the monitored area, they either use GPS or other localization algorithms to determine their next positions within the monitored area. These localization techniques known to be approximate may lead sensor nodes to incorrectly determine their next positions within the monitored area (with an error factor *e*). Therefore, we'll evaluate the impact of on this incorrectness on the number of victims detected within the monitored area.

 ISBN:978-1-61208-113-7

REFERENCES

[1] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, September 2002, vol 13 (2), pp 113-126.

[2] B. Bhattacharya, M. Burmester, Y. Hu, E. Kranakis, and Q. Shi "Optimal Movement of Mobile Sensors for Barrier Coverage of a Planar Region", the 2nd international conference on Combinatorial Optimization and Application, 2008, pp. 5515-5528.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication," Mobicom 2000, pp 56-67.

[4] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks", Infocom 2005, vol. 4, pp. 2302- 2312.

[5] K. Sha, W. Shi, and O. Watkins, "Using Wireless Sensor Networks for Fire Rescue Applications: Requirements and Challenges", The 2006 IEEE International Conference on Electro/information Technology, pp. 239 -244.

[6] I. M. Rekleitis, A. P. New, and H. Choset, "Distributed coverage of unknown/unstructured environments by mobile sensor networks," 3rd International NRL Workshop on Multi-Robot Systems, 2005, pp. 145-155.

[7] M. A. Batalin and G. S. Sukhatme, "Multi-robot dynamic coverage of a planar bounded environment", Technical Report, CRES-03-011, 2003.

[8] N. Heo and PK Varshney. "A distributed self spreading algorithm for mobile wireless sensor networks. Wireless Communications and Networking, 2003, WCNC'2003, vol.3, pp. 1597-1602.

[9] N. Heo and P.K. Varshney, "Energy-efficient deployment of Intelligent Mobile sensor networks", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans , vol. 35(1), pp. 78-92, 2005.

[10] N. Xu, "A Survey of Sensor Network Applications", IEEE Communications Magazine, 2002, vol. 40(8), pp. 102-114.

[11] R. Severino and M. Alves, "Engineering a Search and Rescue Application with a Wireless Sensor Network - based Localization Mechanism", WoWMoM 2007, pp. 1- 4.

[12] S. C. Wong and B. A. MacDonald, "A topological coverage algorithm for mobile robots," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 1685-1690.

[13] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Network," *ACM Mobicom*, August 1999, pp. 174 -185.

[14] http ://*wsnet.gforge.inria.fr*/ inria-00475581, version 1 - 22 Apr 2010.

[15] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces," *INFOCOM* 2003, vol.2, pp. 1293-1303.