Effective Handling of Spreading Events Using Wireless Sensor and Actuator Networks

Wassila Lalouani*, Mohamed Younis**, Miloud Bagaa*** and Nadjib Badache***

*High National School of Computing Science, Algiers, Algeria. Email: w_lalouani@esi.dz
**Dep.t of Computer Science and Elect. Eng., University of Maryland Baltimore County. Email:younis@cs.umbc.edu
***Department of Theories and Computer Engineering, CERIST, Algiers, Algeria. Email: bagaa|badache@mail.cerist.dz

***Abstract*—Wireless sensors and actors networks (WSANs) have the capacity for not only monitoring some phenomena through sensor nodes but also performing appropriate actions. Most of the contemporary WSAN management solutions focus on defining communication path among sensors and actors and on tasking appropriate actors to handle the detected events. In this paper we classify events based on how they evolve over time into continuous and discrete and categorize the WSAN management strategies accordingly. Unlike discrete events, a continuous event spreads quickly and becomes more serious as time passes. Such a characteristic introduces more challenges and motivates a non-conventional management strategies. This paper presents an approach for Sensor-Actuator Coordination for Handling Spreading events (SACHS). SACHS opts to enable the network to respond quickly in order to avoid the event from growing in scope, e.g., prevent a fire from spreading, while reducing the energy overhead due to the coordination messages and due to actor's relocation to the event region. SACHS limits sensor-actor and actor-actor interactions and exploits local sensor-sensor communication to determine the scope of the event, define spots for actors to position at, and schedule the actors' response. The simulation results confirm the performance advantage of SACHS compared to competing schemes.**

***Keywords:* Sensor-Actor networks, Coordination protocols, WSAN management.**

I. Introduction

Recent advances MEMS and communication technologies have enabled the development of highly capable devices that can probe the environment to detect special events and deliver preplanned response. Interconnecting a set of these devices has led to the emergence of wireless sensors and actors networks (WSAN). The WSAN architecture typically has two tiers; the first includes a large number of sensor nodes that are tasks for measuring ambient conditions and communicating their readings over wireless links; the second tier is formed out a relatively fewer actor (actuator) nodes that receive the sensor reports, process the data to make an overall assessment, and implement certain actions when deemed necessary. WSANs are effective in serving applications for which more than area surveillance and environment monitoring are needed. In other word, a WSAN implements a closed control loop in which sensor data are used to define the action. Example of WSAN applications include detecting and putting out fires, combat robotics, target detection and chasing, etc.[1].

Unlike wireless sensors networks (WSNs), where a single base-station is responsible for the network operation and management, in WSANs sensor-sensor, sensor-actuator, and actuator-actuator coordination is required to achieve the overall application objective [2]. In particular, the sensor-actuator coordination ensures the transmission of data reports from sensors that detect an event to an actor(s). Upon receiving the event report, actors coordinate in order to infer the event characteristics and decide on the appropriate actor(s) that are to be assigned the actuation task. Contemporary WSAN management solutions focus on how to associate sensors to actors and how actors decide on what to do and which actuator nodes are to be engaged. The considered design objectives include responsiveness to events, high actor and sensor coverage, and reduced overhead in terms communication and motion related energy consumption.

In this paper we promote a new categorization of WSAN management strategies depending on the nature of the detected event. We classify events based on how they evolve over time into continuous and discrete. A continuous event usually starts in a region within the monitored area and grows in scope over time. Examples of continuous events include hazardous chemicals leakage, Forest fire, and emission of a noxious gas. The seriousness of these events scales spatially as time passes. On the other hand, discrete events do not expand over time and only change location. Intrusion detection and target tracking are examples of discrete events. From a network point of view, the sensors detecting a discrete event vary over time while the sensor population reporting a continuous event grows since the scope of the event widens. No wonder the management strategy of a WSAN would have to differ based on the nature of the events that the network is handling. For discrete events, the actors which will get data reports often change and the inter-actor coordination becomes mostly events handover in many cases. Meanwhile for a continuous event, the number of sensors and actors involved in coordination increases with the size of the event region and consequently the WSAN management becomes challenging.

This paper focuses on the handling of continuous events by first enumerating the technical issues and then devising an efficient network management solution. The scale and growth of the event region make it difficult for actors to gather data from all sensors to assess the seriousness of the situation and determine the boundary of event region, especially for harsh events such as fires. We argue that decentralized sensor-centric solutions will be more appropriate in such a scenario where the proximity of sensors determines the event boundary region without the engagement of actors. Careful assignment of tasks to actors is crucial in order to ensure efficient event reaction. Such coordination opts to engage the fewest actors in covering the event region and is more challenging to achieve for regions with irregular shape. Since in many applications continuous events are hazardous, the task assignment step should factor in the event proprieties to ensure that the actuators will not be damaged. Furthermore, the actors'

978-1-4799-0959-9/14/$31.00 ©2014 IEEE

response to the event should be rapid in order to prevent it from spreading further. Therefore, the assignment of actors should factor in their proximity and avoid redundant coverage in order to conserve resources and be prepared for the possibility of having another simultaneous event.

In this paper we present an approach for Sensor-Actuator Coordination for Handling Spreading events (SACHS). SACHS promotes an efficient WSAN management strategy that suits continuous events. The objective is to respond quickly in order to avoid the event from growing in scope, e.g., prevent a fire from spreading, while reducing the energy overhead due to the coordination messages and due to actor's relocation to the event region. SACHS limits sensor-actor and actor-actor interactions over all networks. Basically, we divide the event handling into three steps, namely, boundary recognition of the event, coverage of the event region, and task assignment to actuators. First, we present a localized sensor-centric algorithm for determining the boundary of the event region using Voronoi diagram. Second, we employ a coverage solution that splits the event polygon, determined during the boundary recognition step, into multiple non-overlapping circles with radius that equals the effective range of the individual actuators. These circles are assigned to actors in the third step while factoring in the actor's energy constraint, the delay in reaching the targeted region and in responding to the event, and the characteristics of the detected event. SACHS is validated through simulation and is shown to outperform competing schemes.

The paper is organized as follows. The next section summarizes related work in the literature and compares SACHS to existing solutions. Section III discusses the system model and states the assumptions. SACHS is described in detail in Section IV. Section V presents the simulation results. Finally section VI concludes the paper and highlights our planned extension for SACHS.

II. Related Work

The design of WSANs has received significant attention from the research community in recent years, with a focus on the sensor-actor and inter-actor coordination [1][2]. The objective of the sensor-actor coordination is to associate an appropriate actor to each sensor for receiving its data and to establish an optimal route for these data to be disseminated to the assigned actor. Z. D. Wu [3] classified the sensor-actor coordination strategies according to whether sensors are assigned specific actors before or after an event occurs and based on the sensor to actor association criteria, e.g., proximity, hop-count, etc., and compared their performance under various application scenarios. On the other hand, the inter-actor coordination opts to identify the best subset of actors to deal with an event subject to some performance goals, e.g., energy conservation, reduced action latency, etc. The main issues considered are how the protocol deals with actor mobility, handles multiple events and implements the coordination procedure, i.e., centralized or distributed [4]. However, these comparative studies are for handling discrete events and do not consider scenarios with continuous events.

Actor mobility is one of the main issues that affect the coordination strategy, specifically how a sensor will know where to send the data and how to determine the actors that take part in responding to an event while reducing overhead and latency. Some work assumes stationary actors and thus limits the problem to simply splitting sensors among actors and to forming suitable routing topology. This can be done before an event takes place[5], which makes the problem even simpler, or upon the detection of an event in order to be adaptive to changes in network state [6].On the other hand, dealing with mobile actors is more challenging given the potentially high overhead due to the frequent location updates. In [7], to overcome the impracticality of fine-grained tracking of actor positions, Voronoi diagram is employed to limit the scope of dissemination of sensor data to actors. Basically, actors form Voronoi polygon based on their positions and inform sensors. Sensor readings are forwarded only within the Voronoi cell it is located within. Sensors further use a Kalman filter to predict the next position of an actuator. However, the signaling overhead associated with the localization and the formation of Voronoi polygons significantly increases with the number of actuators and their displacement. An adaptive publisher-subscriber model is pursued in [8]to form dynamic clusters where an actor publishes its state, e.g., location and event interest, and then sensors subscribe to a specific actor based on its state. SACHS reduces the overhead by designating some sensors to maintain the state of the actors.

Identifying the actors that ought to handle an event is a key function of a WSAN. Such a designation can be simply based on the actor's proximity to the event [5][6][7][8], by planning [9], or through inter-actor coordination [10][11]. Proximity-based actor tasking suits discrete events and can be the byproduct of sensor clustering. Planning is a variant of proximity-based actor selection that factors in the probability for an event to take place in a specific region. For example, Nagi et al., [9]alters the spatial distribution of actors based on the observed frequency of event occurrences. On the other hand, Zeng et al. [10]relies on actors to decide among themselves on which node to respond based on the certainty that an application specific deadline is to be met. Factors that affect the response time include actor availability, e.g., not busy handling another event, action range, motion speed and distance to the event region.

In summary, all published schemes have focused on the association of sensors to actors in terms of reporting and on the efficient dissimilation of sensor data. The actor that gets notified typically responds to the event and solicits the help of additional actuators as needed. The handling of a continuous event that grows rapidly in scope introduces new challenges, as will be pointed out next, and requires a different management strategy from what is proposed in the literature. SACHS opts to fill such a technical gap.

III. System Model and Problem Definition

We consider a WSAN that is formed of numerous sensors and fewer actuator nodes that are spread throughout an area of interest. The sensors are stationary nodes that are equipped with radio transceivers and form a connected topology. We assume that each sensor node is aware of its location and the position of its direct neighbors, e.g., by applying ranging technique and forming a relative coordinate system[12]. Actuator nodes, or simply actors, are more capable nodes that can move. An actor is designed to provide a response to specific events, e.g., extinguish a fire, and has an effective circular range with radius R_a. Although the communication range of actors is longer than that of sensor nodes, the size of the deployment area is assumed to be so large that actors may not have direct links with one another and inter-actor communication may have to be over multi-hop routes that employ both sensor and actuator nodes as relays. We consider a WSAN with homogenous set of actors, i.e., all actors have the same capabilities. All sensors and actors operate on batteries and energy conservation measures are to be applied at the node and network level in order to extend the service time of these nodes. As pointed out earlier, a continuous event grows in scope overtime, and many sensors and actors get involved. Thus for a continuous event, the following issues should addressed:

- *Determining the boundary of the event region*: For a discrete event it is relatively easy to determine where actors should respond since only few sensors provide reports and the event is within their vicinity. In addition, very limited number of actors are involved in coordination and it is reasonable for them to share and correlate the sensor data and assess the event location and dynamics. However, for a continuous event many sensors report and their data reach multiple dispersed actors. In addition, as time passes the volume of the data traffic grows given the spreading of the event, and even more actors become recipients of the sensor reports. Therefore, determining the scope of the event by actors requires continual interaction among them which introduces prohibitive communication overhead given the volume of data in case of continuous events. When inter-actor connectivity is achieved through paths that have sensors as relays, it is not guaranteed that all actors are reachable to one another or at least some actors may not be reachable over dependable routes given the vulnerability of sensors to failure and the high potential of packet drops. Thus, sharing data among actors would not be robust and close coordination among them would not be an effective means for drawing the boundary of the event region and assigning tasks. SACHS explores the use of sensors for determining the scope of the event as we explain in the next section.
- *Covering the event region both effectively and efficiently*: Once the boundary of the event region is defined, multiple actors have to collaborate on responding to the event within such a region. It is important though to optimize the usage of the available actuators in order to be able to cope with any additional events that may take place. Therefore, based on the actuator coverage model, the least number of actors are to be determined. This is known problem that has been extensively studied in the realm of WSN [14].
- *Allocating the optimal set of actors to handle the event*: The goal of this step is to determine the best subset of actors to handle the event subject to actor availability and application constraints. The event characteristics may impose constraints on the schedule of actor operation within the region. For example to extinguish a fire actuators have to start from the boundary of the affected region and progress inward. SACHS assumes that the effect of the event has to be contained and covering the boundary of the area is a high priority. On the other hand, the number of actors that can participate will naturally depend on how many of them are available. Two scenarios may be encountered. The first is when the number of available actors is fewer than what the event region needs to cover the entire region simultaneously. In this scenario the question would be how to schedule the operation of the available actors in order to effectively and rapidly handle the event while minimizing the travel overhead. The nature of the event would have to be factored in the planning process. The second scenario is when the network has actor count that exceeds what the event needs. In this case the optimization objective will be to minimize the travel overhead and response latency.

IV. Detailed SACHS approach

This section first discusses the WSAN architecture and management strategy, and then describes SACHS in detail.

A. WSAN Management and SACHS Overview

As pointed out, published schemes do not suit continuous events and networks in which actors are reachable to each other only through sensor-based routes. SACHS opts to overcome these limitations by pursuing a WSAN management strategy that relies on the sensor nodes rather than inter-actor coordination. To track the position of actors, SACHS devises a two-tier inter-sensor topology that increases the effectiveness of the network operation. Basically, a subset of the sensors are designated as depot nodes that maintain the status of actor nodes. The depots can be picked either by clustering the sensors and tasking each cluster head to act as a depot, by forming a multi-hop connected dominating set designating dominators as depots, or simply by mapping the deployment area to a grid and picking one sensor in each cell as a depot. In all cases, depots should be reachable to each other and each sensor in the network would know where the closest depot is in terms of hop count. If a depot fails, it can be easily replaced by one of its neighbors and the new depot is to inform nearby sensors as well as other depots[13].

A depot as a repository of the state of actors in its vicinity. When an actor moves, it informs the neighboring sensors in the old and new location. Upon getting notified about the change in an actor's state, the sensor informs the closest depot. In addition to the position update, the actor also reports its energy reserve. When an event is detected, SACHS relies on the sensors to collaboratively determine the boundary of the event region. The closest node on the boundary to a depot will report the detected event and the region it is active within. That depot becomes the lead for orchestrating the response to the event. First the lead depot determines the position of actors that would achieve full coverage of the event region. It then multicasts a request for state update to all other depots in the network. The motive for such a request is to collect the most recent state of all actors. The updated state will be used to choose actors for serving the event region. The selection criteria includes the actor's energy reserve and its proximity to the event region. The lead depot will inform the picked actors through the depots that these actors are close to. The three SACHS modules are discussed in the balance of this section.

B. Determining Event Region Boundary

Boundary recognition has been studied in the context of WSNs. Wang et al. [15] classified the published techniques on network boundary detection into three categories, namely, topological, statistical and geometric methods. Topological methods assume that the node positions are unknown and use multiple rounds of message exchange to determine the boundary. This limits their usage in the case of online contour detection for a continuous event given the large number of sensors that are involved in the process. Statistical methods identify interior and boundary nodes by assuming that the node deployment throughout the network follows some known statistical distribution. This category of algorithms involves complex mathematical computations and is not suited for distributed implementation on resource-constrained nodes. The geometric methods assume that the nodes know their relative location within the network, and generally reproduce more accurate boundary recognition than the other two categories. The most important difference between event and network boundary recognition is the responsiveness aspect as the event may evolve quickly to a wider area. Therefore, SACHS pursues a geometric approach where sensors locally determine whether they are inside or outside the event region.

Let S be the set of sensor nodes and N_i be the set of direct neighbors of a sensor s_i. We denote by f_e: $S \rightarrow \{0,1\}$ the function that associates for each sensor $s_i \in S$ a detectability indicator of event "*e*", such that $f_e(s_i)=1$ or 0 when "*e*" is or is not within the sensing range of s_i, respectively. Let S_e be the set of sensors detecting *e*.The aim is to determine the boundary of an event region. Existing coordination solutions rely on actors in defining the scope of the event, i.e., the boundary of the event region. We argue, however, that this is inefficient in the case of continuous events since excessive network overhead will be imposed if all actors are to be notified, especially when actors are not directly connected to each other due to communication range constraints. Therefore,

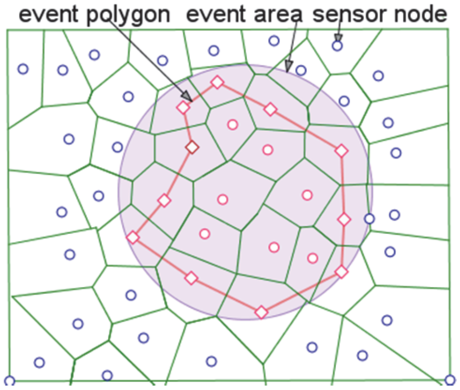

Figure 1: Illustrating how to determine the boundary of the event region using Voronio diagram.

SACHS employs sensors in boundary definition instead. To do so efficiently, SACHS divides the network into n disjoint polytopes $V = \{V_1, \cdots, V_n\}$ using Voronoi diagram, where $n = |S|$. The Voronoi cell of s_i is determined by N_i such that $s_i \in V_i$, $V_i \cap V_j = \varnothing$ for $i \neq j$. Using the properties of the Voronoi diagram, the geometric boundary of the event region can be defined by exploiting the local network connectivity.

Definition 1 The boundary B of an event region is the polygon $p_1, \ldots, p_n$ which separates nodes detecting an event from those that do not.

Lemma1[16]:A point p of a region Q lies on the convex hull of Q iff its Voronoi cell $V_p$is unbounded.

This lemma is used to determine the void in a network or the network periphery. In our case, we exploit this lemma to determine the boundary of an event where the point p of Q lies on the convex hull iff it does not have any Delaunay neighbor $p_1$that detects the event.

Definition 2: A boundary node is a node $x \in S$ such that there exists a Delaunay neighbor $\acute{x} \in N_x$, where$f_e(\acute{x})$=0 and $f_e(x)$=1.

Thus, s_i declares itself as a boundary node if and only if it has at least one Delaunay neighbor that is not detecting the event. In SACHS, each node in S_e performs convexity test to check if it is on the edge of (i.e., a boundary node) or inside the event region. Figure 1 illustrates the idea an example. To determine B, each boundary node s_v sends a message to its right boundary-neighbor, which appends its own ID and forwards the message to the next right boundary-neighbor, and so on. When the message returns to s_v it will include the ID of all boundary nodes, and the hop-distance to its closeted depot. The boundary node with the shortest path to a depot will assume responsibility for reporting the event boundary.

C. Event Coverage

For discrete events, the event region coverage is often a non-issue because the scope of the event is limited and can be handled by one or few actors. However, coverage becomes more important for continuous events since the affected region is large and employing the actors may be constrained by their availability and by the nature of the event. In SACHS determining how to efficiently and effectively cover the event region influences the actor assignment and scheduling step. As mentioned in Section III, a disc coverage model is pursued where an actor is assumed to have a circular action range with radius R_a. The coverage problem is how to place the least number of circles in the event region Ω_e such that every part of Ω_e lies within at least one circle. It was shown in [17] that placing discs on the vertices of a triangular lattice (or, equivalently, at the centers of regular hexagons), is optimal in terms of the number of discs needed to achieve full coverage of a plane. In other words, the distance between the centers of two neighboring circles should be $\sqrt{3}R_a$. While such a coverage problem has been extensively investigated in the context of WSN, the handling of a continuous event in WSANs constrains the solution and motivates the need of non-conventional approach. As most continuous events are serious and often risky, the actors should encircle the event region and progressively move inward in order to contain the event.

SACHS employs an effective solution to determine the positions at which actors should serve in the event region. For each event "*e*", the polygon that defines the region of "*e*" is split into layers that are to be served by actors from the boundary of the event region inward. This model mimics how a continuous event such as a fire is typically served, where it is contained first and then the actors gradually advance towards the center of the event region. Let the points p_1, p_2,…, represent a polygon P that represents the two-dimensional contour determined by boundary recognition step, discussed in the previous subsection. We define the encircled polygon $\acute{P}$ as the connection of points $\acute{p}_i$ such as: $\forall \acute{p} \in \acute{P}, \exists p \in P \mid d(p, \acute{p}) \leq \sqrt{3}R_a$ where $d(\acute{p}, p)$ represents the Euclidian distance between $\acute{p}$and p. The $\sqrt{3}R_a$bound is based on the optimal placement for full coverage of a plane, as mentioned above. The area separating the two polygon P and $\acute{P}$ is named layer L_j. The same steps are repeated to form layers $L_{j+1}, L_{j+2}, \ldots$, etc. The layers are then individually tiled in a descending order with circles of radius R_a that fully covers each layer. The rationale for tiling from the most inner layer outward is due to the fact that the radius of an event region will not typically be multiple of $\sqrt{3}R_a$. If we start from the perimeter inward there may be overlap between the circles in two inner-most layers. On the contrary going outward will limit the overlap and apply the extra coverage to the area around the perimeter, which can make the event handling more robust and mitigate inaccuracy in the definition of the boundary of the event region. Figure 2 illustrates the event coverage procedure.

D. Assignment Tasks to Actors

Given the number and the position of circles determined by the event coverage step, SACHS opts to find the minimum number of actors that can serve the event while minimizing the energy consumption and meeting an application-specific delay bound "Δ". Such a bound often reflects the significance of the event and the rate of growth of the event region. The delay bound is factored in determining the number of actors that ought to be engaged. Upon assigning an actor it moves to the event area and serve layer L_1 and then progresses inward depending its task load. We represent the state of an actor "i"

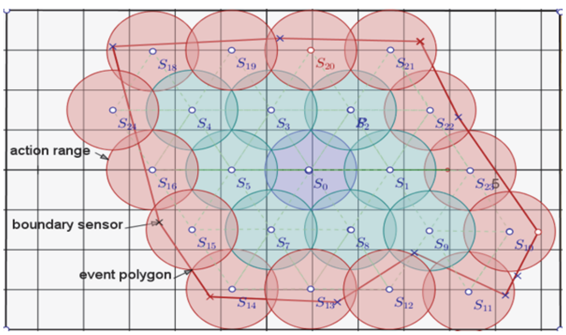

Figure 2: The event region, determined by the boundary recognition step, is divided into layers. Starting with the inner-most layer, a triangular lattice is formed and circles are centered at the vertices of the equilateral triangles. This is repeated layer-by-layer until reaching the boundary.

by its position, residual energy E_i, and availability indicator A_i. An actor informs its closest depot when it moves or sends state update periodically if its position has not changed. When the lead depot receives the event boundary information, it drafts the coverage plan, assigns tasks to actors and then informs the other depots to notify the picked actors in their vicinity. Moving and operating at the center of each circle in the coverage plan constitute a task. The depot opts to assign to each available actor zero or more circles to handle. The following is the optimization formulation of the task assignment problem, where Δ is the delay bound, N_a denotes the number of actors, M is the number of circles to cover the event region, $m_i$is the number of circles assigned to actor i, d_i is the distance between actor i and the event region, x_i is a binary variable set to true if actor i is chosen, and C, ε_t, τ, and ε_a are the travel speed and energy (per unit), and the time and consumed energy for an actor to serve a circle, respectively. The objective is to select a subset of the available actors and assign each of them a number of circles such that the total energy consumed in handling the event is minimized. It is important to note that the travel overhead within the event region is ignored given the closed proximity.

$$\min\left(\sum_{i}^{N_a}(\varepsilon_t * d_i - \varepsilon_a * m_i) \times A_i \times x_i\right)$$

Subject to the following constraints:

$$\Delta \geq \max_{i \leq N_a}(A_i \times x_i \times (d_i \times C + \tau \times m_i)) \ldots\ldots\ldots\ldots. (1)$$

The delay bound for fully handling the event should be met.

$$E_i - \varepsilon_t \times d_i - \varepsilon_a \times m_i > 0 \ldots\ldots\ldots\ldots. (2)$$

Actors should have enough energy to complete assigned tasks.

$$\textstyle\sum_i^{N_a} x_i \leq N_a \ldots\ldots\ldots\ldots. (3)$$

The number of engaged actors is less than what is available.

$$\textstyle\sum_i^{N_a} m_i = M \ldots\ldots\ldots\ldots. (4)$$

All circles are assigned to achieve full event coverage.

```
Sort actors in a descending order of their remaining energy
Cr=0, i = 0;
While (i ≤ Na) and (M > Cr)
  xi= 0;
  find the distance to the nearest circle for each actor i.
  find mi based on equation (2).
  IFmi does not meet the delay constraint (1)
find the max mi that verify the delay constraint
   End IF
   IF (mi>0):
     xi= 1;   Cr= Cr+ mi
   End IF
   i =i+1;
End While
i = 0;   // Have to compromise on delay is M > Cr
While (i ≤ Na) and (xi =0) and (M > Cr)
   calculate miby equation (2)
   IF (mi>0):
     xi= 1;   Cr= Cr+ mi
   End IF
   i =i+1;
End While
```

Figure 3: Pseudo code description of the task assignment step.

To solve this optimization, SACHS tries to engage the least number of actors in order to allow the network to respond to another event that may emerge. In addition, moving few actors will reduce the energy consumed in the relocation and would help in achieving the objective. Therefore, SACHS sorts the actors in a descending order of their residual energy. It then goes through the list assigning the most number of circles to actors while observing (2). If the delay bound (1) is not met, SACHS lowers the load on the picked actors and incrementally engages the next actor(s) on the list until all tasks are allocated. If some tasks remains unallocated, the delay bound is relaxed to engage distant actors. The pseudo code of task assignment procedure of SACHS is shown in Figure 3. The designated actors for the event will then be assigned specific circles on layer L_1 in the event region such that the number of circles they serve in that layer does not exceed their load (m_i). Assigning circles in $L_2$and subsequent layers will follow the same procedure by considering the remaining share of the individual actors until the event region is completely covered.

V. PERFORMANCE VALIDATION

The effectiveness of SACHS is validated through simulation. This section discusses the simulation setup, performance metrics and results.

A. Simualtion Environment and Performance Metrics

We developed a custom simulation environment. In the simulation, we deploy 300 sensors using a uniform random distribution in an area with a varying size to achieve some desired node density. The sensing and action ranges are fixed at 20m and 40m, respectively. The actor moves at a constant speed of 7m/sec. The energy consumed for an actor response is set to 1joul/m^2.The energy of actor displacement is assumed to be1joul/m. These values are used in [7], which we picked as a baseline for performance comparison. An event is generated by choosing a random location to be the center of the event region. From that center, a random number of circles with variant random radii are drawn. The union of these circles constitutes the event region. Sensors within this region can detect the event and run SACHS. Unless varied, the network has 4 actors.

Since there is no published WSAN solution that comprehensively handles continuous events, we compare the performance of specific modules in SACHS to competing schemes. First, the performance of the coordination module is compared to Melodia et al. [7], which present an actor location management scheme to enable efficient geographical routing for sensor-actor communications. The authors opt to reduce delay by exploiting power control at low traffic loads and the spatial diversity of actors is split traffic and mitigate congestion at higher loads. The event region is neither determined nor factored in the sensor reporting, and the inter-actor coordination does not ensure event coverage. Therefore, we have also chosen to compare SACHS with the Savkin et al.protocol [18]. Savkin et al. developed a distributed algorithm to deploy mobile nodes over triangular lattice that spans the entire event region. The nodes starts at the boundary of the region and proceeds inward. This protocol does not deal with the sensor-actor coordination and assumes sufficient actor count to cover the event boundary. In addition, all actors participate in the communication and the reaction.

The simulation opts to capture the performance in terms of the following metrics:

1) *Number of reporting sensors*: denotes the number of sensor that detect and report on an event. This metric reflects the sensor-actor overhead in terms of messaging.
2) *Number of messages*: assesses the traffic volume due to both sensor-sensor and sensor-actor interactions.
3) *Event handling latency*: measured by the time for the last one among the engaged actors to complete its tasks.
4) *Consumed actor energy*: includes the energy for communication, relocation to the event region and in performing the assigned tasks.

B. Experiment Setup and Results

Two sets of experiments have been conducted. This subsection describes the setups and discusses the obtained results. The results reflect the average over 30 different random configurations and stay within %6-%12 of the sample means for a %90 confidence interval. The first experiment focuses on capturing the messaging overhead in comparison with Melodia et al. [7].We track the number of sensors that report to actors about the event (Figure 4). We also track the total number of messages for sensor-sensor and sensor-actor interaction (Figure 5). We fix the number of sensors while varying the network density and the event range since the tracked performance metrics depend on the number of nodes detecting the event and not on the overall sensor population. Controlling the event range is done by picking a random point and using it a center of a circular event region with a specific radius.

After determining the event region boundary, SACHS chooses a boundary node to report the event to the nearest deport. So, only one notification message is sent out of the event region. In Melodia et al., all sensors detecting the event should inform one of the actors which increases the number of reporters and thus the communication energy. As we can see in Figure 4, for SACHS the number of reporters does not depend on the node density or on the event region size. This is not the case for Melodia et al., where the number of reporters grows significantly with the width of the event region and increased node density. This is due to the fact that the number of sensors detecting the event increases.

The energy consumption due to coordination can be captured by considering the total number of messages. For Melodia et al. this reflects messages sent during the sensor-actor coordination, which include the event detection and notifications. For SACHS the messaging overhead is limited tothe event announcement sent to the depot, and boundary nodes' communication to determine the event region. As seen in the Figure 5, the number of messages grows with the node density and the size of the event region because the number of sensors detecting the event increases. The figure clearly demonstrates the advantage of SACHS over the baseline approach with the performance gap significantly widening with increased node density and growth in the event region.

In the second experiment the effectiveness of determining and covering the event region is assessed. Here we fix the number of nodes to 300, the density to 10 and the event range to 30m while varying the number of events. In order to estimate the energy consumed by the actors in serving the events. To assess the latency performance, we measure the time for the last actor to finish while varying the number of events and the size of the event region in terms of the number of circles needed for full coverage. To get the number of

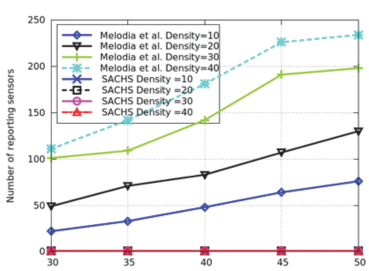

Figure 4: Comparing the number of sensors that send event notifications as the event region and node density grow.

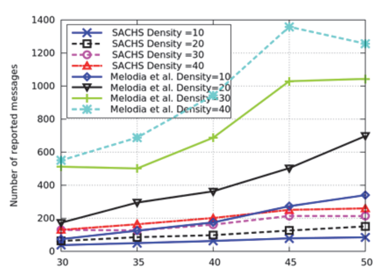

Figure 5: Message overhead comparison for various event region sizes and node densities.

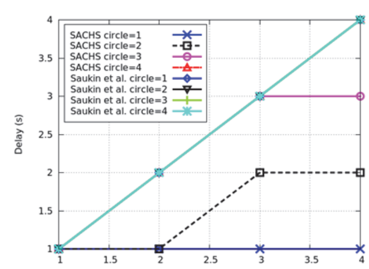

Figure 6:SACHS achieves better latency since it engages only the fewest actors needed for handling the event

circles for an event, we launch the simulation for a random event region and for each execution, we calculate the number of circles in the first (outmost) layer. We focused only on cases with 4 circles or less because we have fixed the number of actors to 4 and the Savkin et al. protocol [14] cannot ensure task accomplishment if there are not sufficient actors. For SACHS, the delay bound for an event is set to 1 second.

In the Savkin et al. protocol, only one event could be handled at a time and the actor count should be sufficient to cover the outer layer. Therefore, we have tracked cased with the same number of circles by event. The results in Figure 6 show that SACHS yields better delay except in the case of one event and when the number of needed actors equals the

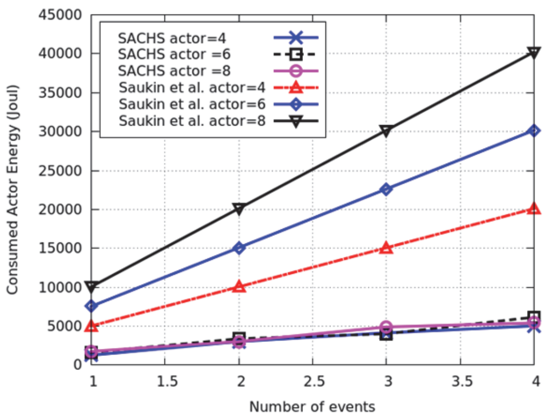

Figure 7: Total energy consumed by actors in handling events.

available count, for which the two protocols gives the same results. In the solution of Saukin et al., the delay is linear in the number of events. Meanwhile SACH makes an appropriate task assignment to actors and thus could handle the events in less time except when all the available actors are engaged in handling each event.

As in SACHS coverage planning and task assignment to actors are performed by a depot node, no energy is consumed in inter-actor coordination. The energy that actors spend in SACHS is for relocating to and performing tasks in the event region. On the other hand, the approach of Saukin et al. requires actors' negotiation for each layer in the event region. Furthermore, their approach mobilizes all actors for each event even if fewer count suffices. As shown in Figure 7, the energy of reaction largely increases with the number of events and the width of the event.

VI. CONCLUSION

A WSAN can be invaluable for multiple applications since it operates autonomously, and can both detect and respond to events without any human intervention. In this paper, we have classifies the WSAN management strategies based on the event evolution over time into discrete or continuous, and outlined new challenges that concern the design of WSANs serving continuous events. We further presented SACHS an approach for Sensor-Actuator Coordination for Handling Spreading events that determines the spatial scope of the event and devises an efficient plan for engaging the actors in covering the event region. The simulation results have demonstrated the effectiveness of SACHS and its performance advantage over competing schemes. Our future work includes extending the approach to handle event regions that have inner spots in which the event is inactive, e.g., a pond within a burning forest, and to factor in other parameters in the task allocation, such as the presence of critical landmarks within the event region for which high priority should be given.

ACKNOWLEDGEMENTS:M. Younis' work is supported by the National Science Foundation (NSF) award # CNS 1018171.

REFERENCES

[1] I. F. Akyildizand I. Kasimoglu, "Wireless sensor and actor networks: research challenges, Ad Hoc Networks, **2**(4), pp. 351–367, 2004.

[2] E. Ruiz-Ibarra, L. Villasenor-Gonzalez, "Cooperation Mechanism Taxonomy for Wireless Sensor and Actor Networks,"*Ad Hoc & Sensor Wireless Networks*,**7**(1-2), pp. 91-113, 2009.

[3] Z. D. Wu, "Modelling and analysis of strategies in the design of WSAN coordination systems," J. Parallel Distrib. Comput. **71**(7), pp. 1055-1064, 2011.

[4] H. Salarian, K.-W. Chin, and F. Naghdy, "Coordination in wireless sensor-actuator networks: A survey," *Journal of Parallel and Distributed Computing*,**72**(7), pp. 856–867, July 2012.

[5] G.A. Shah, M. Bozyigit, and F.B. Hussain, "Cluster-based coordination and routing framework for wireless sensor and actor networks,"*Wireless Communications and Mobile Computing*, **11**(8), pp. 1140–1154, 2009.

[6] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "Communication and coordination in wireless sensor and actor networks," *IEEE Transactions on Mobile Computing*, **6**(10), pp. 1116–1129, 2007.

[7] T. Melodia, D. Pompili, and I. F. Akyildiz, "Handling mobility in wireless sensor and actor networks," *IEEE Transactions on Mobile Computing*,**9**(2), pp. 160–173, 2010.

[8] G. A. Shah, M. Bozyigit, and D. Aksoy, "Adaptive pull-push based event tracking in wireless sensor actor networks," *International Journal on Wireless Information Networks*,**18**(1), pp. 24–38, 2011.

[9] E . C. H. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "A delay-aware reliable event reporting framework for wireless sensor-actuator networks." *Ad Hoc Networks*, **8**(7), pp. 694–707, 2010.

[10]Y. Zeng, C. J. Sreenan, and G. Zheng, "A real-time architecture for automated wireless sensor and actuator networks," *Proc. of the 5th International Conference on Wireless and Mobile Communications (ICWMC'09)*, French Riviera, France, August 2009.

[11]G. A. Shah, M. Bozyigit, O. B. Akan, and B. Baykal, "Real-time coordination and routing in wireless sensor and actor networks," *Proc. of the 6th Int'l Conf. on Next Generation Tele-traffic and Wired/Wireless Advanced Networking (NEW2AN'06)*,St. Petersburg, Russia, May 2006.

[12]A. Youssef, A. Agrawala and M. Younis, "Accurate Anchor-Free Localization in Wireless Sensor Networks," in the *Proceedings of the1stIEEE Workshop on Information Assurance in Wireless Sensor Networks (WSNIA 2005)*, Phoenix, Arizona, April 2005.

[13]A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," *Journal of Computer Communications*, Special Issue on Network Coverage and Routing Schemes for Wireless Sensor Networks, Vol. 30, pp. 2826–2841, 2007.

[14]B. Wang, "Coverage Problems in Sensor Networks: A Survey," ACM Computing Surveys,**43**(4), Article 32, October 2011.

[15]Y. Wang, J. Gao, and J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," *Proc. of 12thAnnual Int'l Conf. on Mobile Comp. and Networking (MobiCom)*,Los Angeles, CA, Sept 2006.

[16]F. Aurenhammer, R. Klein, D.-T. Lee,*Voronoi Diagrams and Delaunay Triangulations*. World Scientific, ISBN 978-981-4447-63-8, 2013.

[17]R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, pp. 665–671, 1939.

[18]A.V. Savkin, Xi Zhiyu,and H.T. Nguyen,"An Algorithm of Decentralized Encircling Coverage and Termination of a Moving Deformable Region by Mobile Robotic Sensor/Actuator Networks,"*Proc. of the 9th Asian Control Conference (ASCC)*, Istanbul, Turkey, June 2013.