# Energy-Efficient Data Collection in Heterogeneous Wireless Sensor and Actor Networks

Waseem Abbas, Hassan Jaleel, and Magnus Egerstedt

***Abstract*—In this paper, we address the issue of activity scheduling of sensors in heterogeneous wireless sensor and actor networks (WSANs), thereby proposing an energy-efficient data collection scheme in such networks. In order to extend the lifetime of heterogeneous WSANs, sensors are activated and deactivated under certain constraints throughout the network operations. Here, we propose a coordination framework in which actors exchange information with each other and decide about the availability of redundant sensors that are eventually deactivated to save energy. In particular, let there be $r$ different types of sensors with each sensor observing a particular sensing parameter. Under the initial deployment of sensors and actors within some field of observation, if an actor $v$ receives information regarding $k$ different sensing parameters, either directly from sensors or through other actors, then our scheme determines a small subset of sensors that are sufficient to provide information regarding the same $k$ sensing parameters to $v$.**

## I. Introduction

Wireless sensor and actor networks (WSANs) provide an effective solution to the distributed sensing and response related problems. In such networks, information gathered by the sensing nodes is made available to the actor nodes through a wireless medium that utilize this information to make decisions and act upon the environment. In comparison to the wireless sensor networks (WSNs), which are generally designed to observe the environment and then transfer these observations to the sink nodes, WSANs possess the capability of not only observing but also affecting the environment by using the observations [1].

The components in WSANs are categorized into two major classes: sensors and actors. Sensors provide a distributed sensing infrastructure, and are typically inexpensive, low-power devices with limited computational and communication capabilities [1]. Owing to these properties, sensors are generally deployed in greater numbers. Actors, on the other hand are more sophisticated and resource-rich nodes with longer battery life, higher processing skills, and transmission powers. They are capable of processing data obtained from the sensors, and then taking appropriate actions. Robots and unmanned ground or aerial vehicles are the examples of actors. Typically it is assumed that the number of actors in a network is much smaller than the sensors [2].

Heterogeneity emerges as an important property of WSANs in which sensors with a varying set of sensing and transmission capabilities are deployed within some field of interest. For instance, for the purpose of environment modeling, a set of temperature, air flow, and pressure sensors may be deployed in the field to observe various aspects of the climate. Moreover, an actor present in the field may not have access to all sorts of measurements by the sensors. Thus, actors rely on coordination and communication with each other to acquire complete information under the automated architecture of WSANs [1]. At the same time, managing the activity of sensor nodes through efficient activity scheduling mechanisms is imperative for a longer lifetime of the sensor network as they have limited power resources.

In this paper, we address the issue of energy-efficient information gathering in a heterogeneous WSAN. A scheme is proposed in which actors utilize coordination among themselves to deactivate a large portion of originally deployed sensors of various types to preserve sensors' power resources. Under the initial deployment of sensors, if an actor $v$ receives a particular type of data either directly from a sensor, or by interacting with a neighbor actor, then our objective is to activate the minimum number of sensors that ensure the availability of the data to all such actors. We formulate this problem in graph-theoretic terms, thereby providing solutions using graph-coloring and graph-domination related concepts (e.g., [3], [4]).

The issue of designing energy-efficient strategies for longer lifetimes of the networks, while preserving properties like connectivity and coverage, has been extensively studied in the literature for the case of pure WSNs (e.g., [2], [4]). Some of the main activity scheduling schemes for WSNs include the geographical adaptive fidelity (GAF) algorithms; the adaptive self-configuring sensor network topologies (ASCENT) mechanism; the probing environment and adaptive sleeping (PEAS) algorithm; and SPAN, which is a distributed, randomized algorithm proposed to increases the lifetime of the sensor network. A nice survey of these algorithms along with references is given in [2].

However, all of the above schemes are primarily designed for WSNs consisting of sensor nodes only. In the context of WSAN framework, actors are the information processing and response stations that are distributed throughout the network. Actors are resource-rich nodes and actor-actor coordination can be used to develop activity scheduling mechanisms for the sensor nodes of various types. We utilize interactions between actors and heterogeneity among sensors to design an energy-efficient scheduling scheme while ensuring that even after the deactivation of a certain number of sensors, actors continue to obtain the same information as they were acquiring when all sensors were on.

W. Abbas, H. Jaleel, and M. Egerstedt are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA wabbas@gatech.edu, hjaleel3@gatech.edu, magnus@gatech.edu

## II. System Description

Let there be $r$ different types of sensors. Sensors of each type are deployed at random in some domain $\mathcal{A} \subset \mathbb{R}^2$ such that the location of each sensor is independent of other sensors' locations. Such a deployment of sensors can be modeled as a stationary Poisson point process with constant intensity[1] [5]. All of the sensors have a footprint of the form of a closed ball of some radius that depends on the type of the sensor. Let us say that each sensor belongs to one of the types in the set $\mathbf{r} = \{1, 2, \cdots, r\}$, then the deployment of sensors of each type can be modelled by a stationary Poisson point process with intensity $\lambda_i$ where $i \in \mathbf{r}$. Further, we use the following notations:

$r$ : total number of sensor types.

$\lambda_i$ : expected number of sensors of type $i$ per unit area modelled as a stationary Poisson point process.

$\Delta_i$ : radius of the footprint of a sensor of type $i$.

$\alpha_i$ : area of the sensor footprint of type $i$ ($\alpha_i = \pi\Delta_i^2$).

Meanwhile, actors (robots), which are the resourceful nodes within the network capable of performing different tasks after receiving data from various sensors, are also distributed at random and independent of other actors' locations. Thus, actors can also be modeled by a stationary Poisson point process with intensity $\rho$. As in the automated architecture of WSANs, actors coordinate with each other by communicating and exchanging information. An actor interacts with all actors lying within the distance $\Delta_a$ from it. This gives an interaction network that can be modelled by delta-disk proximity graphs. We will use the following notations throughout the paper.

$\rho$ : expected number of actors per unit area.

$\Delta_a$ : communication range of an actor
(radius of the footprint of an actor).

$\alpha_a$ : area of the actor's foot print ($\alpha_a = \pi\Delta_a^2$).

As mentioned earlier, typically the number of actors is much smaller than the number of sensors, thus, $\rho < \lambda_i$. Moreover, actors have higher transmission ranges, i.e., $\Delta_a$ is usually higher than $\Delta_i$. A sensor which is in an active mode (on state) transmits its data to an actor lying within its footprint. Sensors do not communicate with each other, whereas actors transmit and receive information from other actors as well as sensors. Every actor performs tasks that require data from every sensor type, i.e., an actor needs to have information of all $r$ sensing parameters. We consider that the spatial gradients of the sensing modalities observed by sensors are not too large within the field of observation, i.e., there are no abrupt variations in the sensing modalities throughout the field of interest. Therefore, an actor can receive information regarding the $i^{th}$ ($i \in \mathbf{r}$) sensing parameter either directly from the sensor if there exits a sensor of type $i$ within $\Delta_i$ distance from the actor, or through one of the neighboring actors which is directly receiving data from a sensor of type $i$.

Under this set-up, the probability of an actor to receive information regarding all different sensing parameters in $\mathbf{r}$, either directly from sensors, or through adjacent actors depends on various factors including $\lambda_i$ and $\Delta_i$. Thus, for each $i \in \mathbf{r}$, increasing $\lambda_i$ (number of sensors of type $i$) and $\Delta_i$ (transmission range of a sensor of type $i$), will increase the number of actors receiving information of all $r$ different sensing parameters, but only at an additional cost. However, it is observed that owing to the random deployment of sensors with $\lambda_i$ intensities, there exist redundancy within the network in the sense that a lot more sensors are on than required. We can get rid of this redundancy by turning off the redundant sensors for an energy-efficient operation of the system.

Thus, our *objective* is to develop a systematic scheme to turn off the maximal number of redundant sensors of all types in a distributed manner while ensuring the following: if an actor $v$, or one of its neighbor actors (actors directly connected to $v$) are lying within the footprint of a sensor of type $i$ in the initial deployment (when all sensors are on), then the same should be true even when the redundant sensors are turned off.

## III. A Graph-theoretic Model

The above problem can be investigated in graph theoretic terms. The network of actors (robots) can be modeled by a graph $G(V, E)$ where the vertex set $V$ represent actors and the edges in the edge set $E$ correspond to the interactions among them. Heterogeneity which exists within the system in the form of sensors of various types can be modeled using the *graph coloring* notion. Graph coloring is an assignment of colors (labels) from a coloring set to the vertices of the graph subject to certain constraints. In [3] and [6] heterogeneous multiagent systems are modeled and various properties of such systems are studied using the graph coloring approach.

Here, the vertices in the graph (representing an actor network) are colored in accordance to the types of sensors directly transmitting data to the corresponding actors. Since $r$ different types of sensors are available within the system, the coloring set is $\mathbf{r} = \{1, 2, \cdots, r\}$. Vertices in the graph $G$ are then assigned labels according to the following labeling function:

$$f : V \longrightarrow 2^{\mathbf{r}}$$

where $2^{\mathbf{r}}$ is the set of all subsets of $r$.

If there exists at least one sensor of type $i \in \mathbf{r}$ within a distance $\Delta_i$ from an actor $v$, then the corresponding vertex in $G$ will be assigned the label (color) $i$. Thus,

$$f(v) = \left\{ \begin{array}{ll} i \in \mathbf{r} \mid & \text{at least one sensor of type } i \text{ exists} \\ & \text{within } \Delta_i \text{ distance from } v. \end{array} \right\} \quad (1)$$

An actor receiving data directly from a sensor of type $i$ exchanges it with the neighboring actors as actors are

[1] Expected number of sensors in a unit area.

interacting and exchanging information with each other. Let us define the *open neighborhood* of a vertex $v$, denoted by $\mathcal{N}(v)$, as the set of vertices in $G$ adjacent to $v$. Similarly, the *closed neighborhood* of $v$, denoted by $\mathcal{N}[v]$, is $\mathcal{N}(v) \cup \{v\}$. Further, we define $\mathcal{F}(v)$ as the set of colors from $\mathbf{r}$ a vertex $v$ can find in its closed neighborhood, i.e.,

$$\mathcal{F}(v) = \bigcup_{u \in \mathcal{N}[v]} f(u)$$

Vertices in the graph $G$ are actors and are labeled in accordance with the types of sensors that directly transmit data to actors. For instance, actor $u$ lies within the footprints of sensors of type 1 and 2, the corresponding vertex $u$ in $G$ is assigned labels 1 and 2, i.e., $f(u) = \{1, 2\}$. Also, $u$ is directly connected to $v$ and $x$ which have labels 3 and 2 respectively, thus, $\mathcal{F}(u) = \mathbf{r} = \{1, 2, 3\}$.

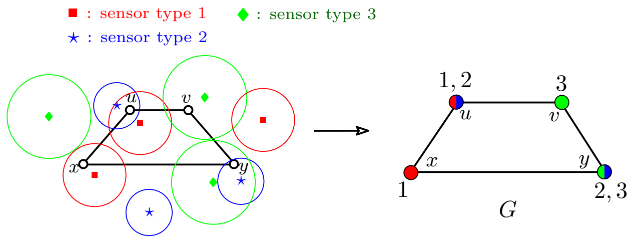

Fig. 1. Three different types of sensors are distributed. The vertices in the graph $G$ (representing an actor network) are assigned colors (labels) from the set $\mathbf{r} = \{1, 2, 3\}$ in accordance with the sensor types directly available to the corresponding actors.

### A. Objective

Sensors of each type $i \in \mathbf{r}$ are distributed at random and independent of each other with intensity $\lambda_i$. Thus, the colors (labels) assigned to the vertices in the above mentioned graph-theoretic model directly depend on the distribution of sensors. We call the labeling of vertices in $G$ due to the initial random deployment of sensors as $\mathcal{L}_{ini}(G)$. Under the labeling $\mathcal{L}_{ini}(G)$, a vertex $v$ in $G$ is assigned labels $f_{ini}(v)$, and $\mathcal{F}_{ini}(v)$ is the set of labels available in the closed neighborhood of $v$. Thus, our goal is to develop a systematic scheme to obtain a new labeling of $G$, i.e., $\mathcal{L}_{new}(G)$ from $\mathcal{L}_{ini}(G)$ by getting rid of some of the labels (redundant labels) assigned to the vertices while ensuring that under this new labeling (which is derived from $\mathcal{L}_{ini}(G)$), every vertex finds exactly the same set of labels in its closed neighborhood as in $\mathcal{L}_{ini}(G)$. More precisely, for every vertex $v$ in $G$, we want to find $f_{new}(v) \subseteq f_{ini}(v)$ in a distributed manner such that $\mathcal{F}_{new}(v) = \mathcal{F}_{ini}(v)$. Since the labels assigned to the vertices correspond to the sensors transmitting data to the actors, getting rid of the labels mean that the corresponding sensors can be turned off leading towards an energy-efficient operation of the sensor and actor network.

## IV. Energy-Efficient Data Collection Scheme

In this section, we present a scheme to turn off the redundant sensors for energy efficiency. Our proposed scheme utilizes both randomness in the deployment of sensors of various types within the region $\mathcal{A} \subset \mathbb{R}^2$, and the coordination among actors to determine and resolve the redundancy existing within the sensor network. Every sensor is considered to have two modes, *active* (on) mode and *de-active* (off) mode. A sensor transmits its data to the actors lying within sensor's footprint only in the active mode. Our scheme consists of the following rounds:

### A. Randomization

Sensors of each type $i \in \mathbf{r}$ are deployed randomly and independently of each other with intensity $\lambda'_i$. At time $t = 0$, each sensor enters into the active mode with some probability $p > 0$. Thus, the effective intensity of sensors of type $i$ will be $\lambda_i = p\lambda'_i$. In order to keep the same expected on time for all the sensors during the overall lifetime of the system, this step is repeated after fixed intervals $t_\delta$. The sensors that become active as a result of this step start transmitting their data to the actors lying within the footprints of these sensors. Every actor maintains a list of the types of sensors it is receiving information from, i.e., $f(v)$ as defined in (1).

### B. Determination of Redundant Sensors

Once $f(v)$ is determined by every actor $v$, the next key step is the exchange of $f(v)$ by every $v$ with its neighbors to determine the existence of redundant sensors within the footprint of an actor. Once determined, these redundant sensors will be de-activated through a de-activating message by the actor to the sensors. The graph-theoretic model of the system introduced in Section III will be used here for the purpose of determining redundant sensors. Every vertex in a graph $G$ modeling the actor network is assigned labels $f(v)$.

Our goal is to obtain for every $v \in V$, a subset $s(v) \subseteq f(v)$ with the minimum cardinality such that $\bigcup_{u \in \mathcal{N}[v]} s(u) = \mathcal{F}(v)$. We deal with this problem individually for each $i \in \mathbf{r}$. Let $V_i$ be the set of vertices having label $i$, i.e.,

$$V_i = \{v \in V : \ i \in f(v)\}$$

Also, let $\tilde{V_i}$ be the set of vertices that have at least one vertex with a label $i$ in their closed neighborhoods, i.e.,

$$\tilde{V_i} = \{v \in V : \ i \in \mathcal{F}(v)\}$$

It is to be noted that $\tilde{V_i} = \bigcup_{v \in V_i} \mathcal{N}[v]$. Thus, for the label $i$, we need to find a subset $S_i \subseteq V_i$ with the minimum cardinality such that $\bigcup_{s \in S_i} \mathcal{N}[s] = \tilde{V_i}$.

In a special case when every vertex is assigned the label $i$, i.e., $V_i = V$, this problem becomes a well known problem in graph theory known as the *dominating set* problem.

**Definition 4.1:** (Dominating Set) A subset $S \subseteq V$ is a dominating set whenever every $v \in V$ either lies in $S$, or is adjacent to at least a vertex in $S$.
Simply, $S$ is a dominating set whenever $\bigcup_{s \in S} \mathcal{N}[s] = V$. For our case, when $V_i$ may not be equal to $V$, we define the *restricted dominating set* as

**Definition 4.2:** (Restricted Dominating Set) Let $V_i \subseteq V$, a subset $S_i \subseteq V_i$ is a restricted dominating set with respect to $V_i$ whenever $\bigcup\limits_{s \in S_i} \mathcal{N}[s] = \bigcup\limits_{v \in V_i} \mathcal{N}[v]$.

An example is shown in Fig. 2.

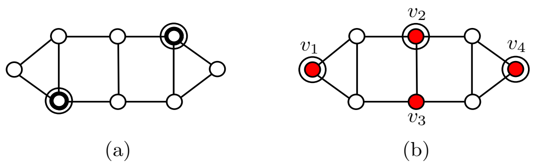

Fig. 2. (a) The circled vertices form a dominating set of the graph. (b) If $V_1 = \{v_1, v_2, v_3, v_4\}$, then a restricted dominating set with respect to $V_1$ consist of the circled vertices, i.e., $S_1 = \{v_1, v_2, v_4\}$.

*Computation of a Restricted Dominating Set $S_i$:*

The problem of finding a minimum dominating set is NP-hard (e.g., [10]) leading to the fact that finding a minimum restricted dominating set is also NP-hard. Thus, finding efficient algorithms for the approximate solutions has been an active area of research. The simplest approach is the *greedy approach* in which a vertex covering the maximum number of uncovered vertices is added into a dominating set at each step [7]. The greedy algorithm achieves an approximation ratio of $\ln \Lambda$ in $O(n)$ time, where $\Lambda$ and $n$ are the maximum degree[2] and total number of vertices in the graph respectively [7]. A *distributed* version of the greedy algorithm is presented in [8], [9]. Interestingly, it is shown in [10] that unless P≈NP, the $\ln \Lambda$-approximation ratio of the simple greedy approach is optimal (upto lower order terms). Therefore, the problem of finding a small *restricted dominating set* with respect to $V_i \subseteq V$ can be solved using the simplest distributed greedy approach. Below, we present a distributed greedy algorithm adapted from [8] for finding a restricted dominating set with respect to $V_i$. Unlike [8] where every $v \in V$ executes a greedy routine, here the algorithm is executed only by the vertices in $V_i$.

Let us define a *dominated node* as the one whose closed neighborhood contains at least one vertex form the restricted dominating set. A vertex is said to be *undominated* if it is not a dominated one. Also let $\mathcal{U}(v)$ = number of undominated nodes in $\mathcal{N}[v]$.

**Algorithm I:** Restricting Dominating Set w.r.t $V_i \subseteq V$

```
$v \in V_i$
**While** $\mathcal{U}(v) > 0$ **do**
    **if** $\mathcal{U}(v)$ is largest among the vertices in $V_i$ that
    are at a distance of at most 2 from $v$ (ties are
    resolved by ID's) **then**
        $v$ joins a restricted dominating set $S_i$
    **end if**
**end while**
```

In the case of $V_i = V$, Algorithm I becomes the original distributed greedy algorithm given in [8] where it is shown that the algorithm returns a dominating set of size that is at most $(\ln \Lambda + 2)$ of the optimal in $O(n)$ time. Thus, using the similar approach as in [8], we get the following:

**Proposition 4.1:** For a given $V_i \subseteq V$, if $S^*$ is a minimum restricted dominating set with respect to $V_i$, then Algorithm I returns a restricted dominating set with respect to $V_i$ of size at most $(\ln \Lambda + 2) \cdot |S^*|$ in $O(n)$ time. Here, $\Lambda$ is the maximum degree of a graph.

It is to be mentioned here that there are numerous other approaches to find small-sized dominating sets (e.g., [11], [12]), but we discussed the greedy algorithm here because of its simplicity and good approximation ratio. The problem of finding a smaller restricted dominating set can be solved by any of the approximation schemes for minimum dominating set.

*Redundant Sensors:*

For our original problem, a restricted dominating set with respect to $V_i$ is computed by the actor network for each sensor type $i \in \mathbf{r}$. Thus, a subset of labels $s(v) \subseteq f(v)$ is determined for each $v \in V$, meaning that the vertices can get rid of some of the labels initially assigned to them while preserving the required condition $\bigcup\limits_{u \in \mathcal{N}[v]} s(u) = \bigcup\limits_{u \in \mathcal{N}[v]} f(u)$, $\forall v \in V$. The assignment of label $i$ to a vertex $v$ represented that the corresponding actor $v$ lies in the footprint of a sensor(s) of type $i$ . Thus, if $i \notin s(v)$, then the sensor(s) of type $i$ is redundant for the actor $v$, and deactivation of the sensor(s) will not affect the data collection by the actor $v$. This leads us to the next step in our scheme.

### C. Deactivation of Redundant Sensors

As previously, we deal with the sensors of each type $i \in \mathbf{r}$ independently of each other in this step. $S_i$ is the set of restricted dominating actors with respect to $V_i$ as computed in Section IV-B. Thus, every $v \in S_i$ needs to have a sensor of type $i$ transmitting data to $v$ directly, i.e., $v$ should be lying in the footprint of an active sensor of type $i$. In fact, only the actors in $S_i$ are the ones that need to receive data directly from a sensor of type $i$. Moreover, it is sufficient for $v \in S_i$ to receive data from only one such sensor. Thus, every $v \in S_i$ broadcasts a deactivating message to all the $i$-type sensors in $v$'s footprint except for a single sensor (of type $i$) which receives an activating signal from $v$. The sensor receiving an activating message can be the one that is nearest to $v$. On the other hand, every $u \in (V - S_i)$ also broadcasts a deactivating signal to *all* the sensors of type $i$ in $u$'s footprint as $u$ does not need to receive directly from an $i$ type sensor. Sensors not receiving any of the activating or deactivating signal are the ones that do not lie within any actor's footprint, and are deactivated eventually. Also, an activating signal has a greater priority, thus, a sensor receiving an activating as well as a deactivating message will become activated. This procedure will be performed for each type ($i \in \mathbf{r}$) of sensors.

After a fixed interval $t_\delta$, all three steps (randomization, determination of redundant sensors, and deactivation of re-

[2] degree of a vertex $v$ is the cardinality of $\mathcal{N}(v)$.

dundant sensors) are repeated. An example of the scheme is discussed in Section VI.

## V. Analysis of the Random Distribution of Sensors

In WSANs, one way to characterize the random deployment of sensors of various types with $\lambda_i$ intensities is to determine the number of actors that receive all types of data either directly from sensors, or by interacting with other actors. An exceeding percentage of such actors is highly desirable as it will imply an extended data access to the actors. In order to estimate this number, we proceed by introducing the following terms:

**Definition 5.1:** In a colored graph $G$ with $\mathbf{r} = \{1, 2, \cdots, r\}$ colors, A vertex $v$ is said to be *completely colored* whenever

$$\mathcal{F}(v) = \mathbf{r}$$

In other words, a vertex $v$ is completely colored whenever it can find every color in the coloring set $\mathbf{r}$ in its closed neighborhood. Similarly, in terms of the actor network, we say that an actor $v$ is *completely covered* whenever $\mathcal{F}(v) = \mathbf{r}$

We are interested in finding the probability of a vertex $v$ being completely colored under the system model described in Section II. It is to be recalled that the deployment of sensors of type $i$ is modeled as a stationary spatial Poisson point process with constant intensity $\lambda_i$. The probability of having $k$ sensors in an area $A$ is then given by (e.g., [5]).

$$P_k = \frac{(\lambda_i A)^k e^{-\lambda_i A}}{k!} \tag{2}$$

**Theorem 5.1:** For an actor $v$ in the wireless sensor and actor network described in Section II, the probability of the existence of an actor $u \in \mathcal{N}[v]$ such that $u$ lies in the footprint of at least one sensor of type $i$ for a given $i \in \mathbf{r}$ is

$$P\left(i \in \mathcal{F}(v)\right) \;=\; 1 - e^{-\left[\lambda_i \alpha_i + \rho\alpha_a(1 - e^{-\lambda_i \alpha_i})\right]} \tag{3}$$

where $\lambda_i$ and $\alpha_i$ are the intensity and the area of the footprint of sensor of type $i$ respectively, whereas $\rho$ and $\alpha_a$ are the intensity and the area of footprint of actor respectively.

*Proof:*

$$P(i \in \mathcal{F}(v)) = 1 \;-\; P(i \notin \mathcal{F}(v))$$

$$P(i \notin \mathcal{F}(v)) = P(i \notin f(v)) \cdot \prod_{u \in \mathcal{N}(v)} P(i \notin f(u)) \tag{4}$$

Here, $P(i \notin f(v))$ is the probability that the label $i$ is not assigned to the actor $v$. After inserting $k = 0$ and $A = \alpha_i$ in (2), we get $P(i \notin f(v)) = e^{-\lambda_i \alpha_i}$.

Similarly, $\prod\limits_{u \in \mathcal{N}(v)} P(i \notin f(u))$ in (4) is the probability that none of the actors in the open neighborhood of $v$ are assigned label $i$. We utilize (2) and standard results from stochastic geometry [5] to get

$$\begin{aligned}\prod_{u \in \mathcal{N}(v)} P(i \notin f(u)) &= \sum_{n=0}^{\infty} P\left(\mid \mathcal{N}(v) \mid= n\right) \cdot \left[P\left(i \notin f(u)\right)\right]^n \\ &= \sum_{n=0}^{\infty} \frac{(\rho\alpha_a)^n \; e^{(-\rho\alpha_a)}}{n!} \; (e^{-\lambda_i \alpha_i})^n \\ &= (e^{-\rho\alpha_a})(e^{\rho\alpha_a e^{-\lambda_i \alpha_i}})\end{aligned}$$

Inserting these values in (4) gives the following after some simplification.

$$P(i \notin \mathcal{F}(v)) \;=\; e^{-\left[\lambda_i a_i + \rho a'(1 - e^{-\lambda_i a_i})\right]}$$

Thus,

$$P\left(i \in \mathcal{F}(v)\right) \;=\; 1 - e^{-\left[\lambda_i \alpha_i + \rho\alpha_a(1 - e^{-\lambda_i \alpha_i})\right]}$$

■

Using the fact that the sensors of each type are deployed independent of each other, and therefore, colors are assigned to the vertices in a graph representing an actor network independent of each other, we deduce the following useful result.

**Corollary 5.2:** The probability of an actor $v$ to be completely covered in the wireless sensor and actor network described in Section II is

$$\begin{aligned}P\left(\mathcal{F}(v) = \mathbf{r}\right) \;&=\; \prod_{i=1}^{|\mathbf{r}|} P\left(i \in \mathcal{F}(v))\right) \\ &=\; \prod_{i=1}^{|\mathbf{r}|} \left[1 - e^{-\left[\lambda_i a_i + \rho a'(1 - e^{-\lambda_i a_i})\right]}\right]\end{aligned} \tag{5}$$

■

We observe that under the random distribution of sensors, $P\left(\mathcal{F}(v) = \mathbf{r}\right)$ can be improved by increasing $\lambda_i$ and $\Delta_i$ for each $i \in \mathbf{r}$. However, increasing $\lambda_i$ means increasing the number of sensors of type $i$, which is costly. Likewise, a higher $\Delta_i$ means sensors need to transmit farther requiring extra power. Thus, we aim to achieve a higher $P\left(\mathcal{F}(v) = \mathbf{r}\right)$ in an economical way (i.e., by keeping the number of active sensors low as well as smaller $\Delta_i$). In fact, the energy-efficient data collection scheme described in Section IV achieves this goal. The underlying objective is to determine all such sensors that are redundant in the sense that their deactivation will not effect the availability of data to the actors, and then eventually turn them off. Let $X$ be the set of completely covered actors as a result of the random deployment of sensors of type $i$, $\forall i \in \mathbf{r}$. Using Proposition 4.1, we deduce that if $S_i^*$ is the minimum number of sensors of type $i$ that need to be activated to ensure that each actor in $X$ is completely covered, then using simple distributed greedy algorithm (Algorithm I), our scheme makes every actor in $X$ completely covered by activating at most $(\ln \Lambda + 2) \cdot |S_i^*|$ sensors of type $i$ which is significantly smaller than the original number of deployed sensors of type $i$.

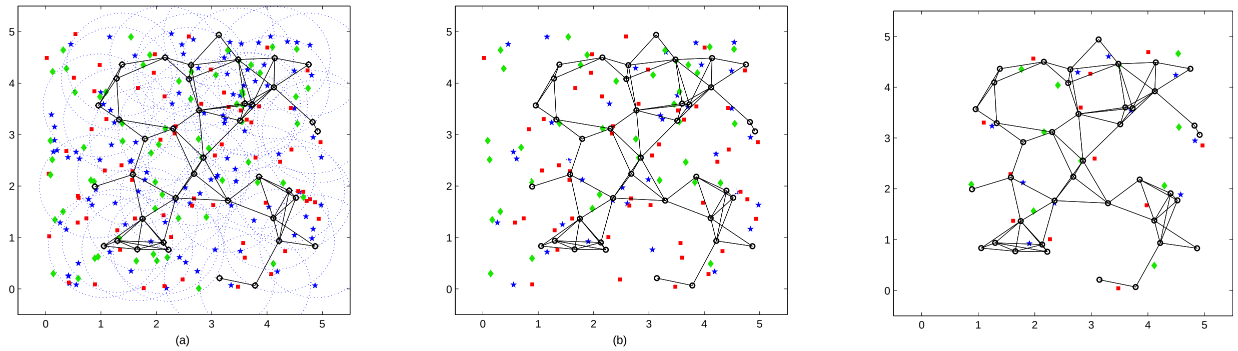

Fig. 3. (a) Sensors of three types $1, 2$, and $3$ (represented as $\Box$, $\star$, and $\Diamond$ respectively) are distributed at random along with an actor network represented by a graph. (b) Active sensors after the randomization step are shown. (c) All the redundant sensors are de-activated.

## VI. Example

Here, we present an example to illustrate the scheme discussed in Section IV. Consider a region with an area $\mathcal{A}$ in which sensors of three different types are distributed at random and independent of each other. Every sensor belongs to one of the types in $\mathbf{r} = \{1, 2, 3\}$. The distribution of sensors of each type $i \in \mathbf{r}$ is modeled as a Poisson point process with intensity $\lambda_i$. The radius of the footprint of a sensor of type $i$ is $\Delta_i$. For our example, $\lambda_1' = 3$, $\Delta_1 = 0.4$; $\lambda_2' = 4$, $\Delta_2 = 0.3$; and $\lambda_3' = 3$, $\Delta_3 = 0.5$. The actors are also distributed at random and independent of each other with intensity $\rho = 1.5$. Every actor has a footprint of radius $\Delta_a = 1$. An actor $v$ interacts with all the actors lying within $v$'s footprint. A graph representing interactions among actors along with the distribution of sensors is shown in Fig. 3.

In the *randomization* phase, a sensor of type $i$ becomes activated with a probability $p_i$. Here, $p_1 = 0.6$, $p_2 = 0.4$, and $p_3 = 0.5$, therefore, $\lambda_1 = 1.8$, $\lambda_2 = 1.6$, and $\lambda_3 = 1.5$. The activated sensors after the randomization phase are shown in Fig. 3(b).

Next step is the *determination of redundant sensors* using the restricted dominating set approach discussed in Section IV-B. All the redundant sensors are then deactivated by the actors as shown in Fig. 3(c) leading to an energy-efficient data collection scheme. The number of active sensors after different phases is given in Table I.

TABLE I

Number of active sensors after various phases.

| | initially deployed | after randomization | after the deactivation of redundant sensors |
|---|---|---|---|
| Type 1 | 69 | 47 | 12 |
| Type 2 | 104 | 35 | 10 |
| Type 3 | 64 | 36 | 11 |

For given $\Delta_i$ and $\lambda_i$ where $i \in \{1, 2, 3\}$, the probability of an actor $v$ being completely covered is 0.854, i.e., $P(\mathcal{F}(v) = \{1, 2, 3\}) = 0.854$. There are 40 actors in total, out of which 36 (90%) are completely covered. Recall that in a completely covered actor $v$, for each $i \in \mathbf{r}$, there exists an actor $u \in \mathcal{N}[v]$ such that $u$ lies within the footprint of at least one sensor of type $i$. Also note that for each sensor type, more than two-third of the sensors that were active after the randomization phase are deactivated in the final step.

## VII. Conclusions

In this paper, we addressed the issue of energy-efficient operations of heterogeneous WSANs. Redundancy among the sensors of various types was explored using notions from the domination in graphs. Sensors that were redundant in the sense that their deactivation did not effect the availability of data to the actors were determined and eventually turned off to save energy. A number of simulations performed with a wide range of $\lambda_i$ and $\Delta_i$ showed that typically more than two-third of the sensors can be deactivated without compromising the availability of data to the actors.

## References

[1] I. Akyildiz and I. Kasimoglu, “Wireless sensor and actor networks: research challenges,” *Ad Hoc Networks*, vol. 2, pp. 351–367, 2004.

[2] I. Akyildiz and M. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, New York, 2010.

[3] W. Abbas and M. Egerstedt, “Distribution of agents with multiple capabilities in heterogeneous multiagent networks – A graph theoretic view,” In *Proc. IEEE Conf. Decision and Control*, pp. 290–295, Maui, HI, 2012.

[4] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.

[5] D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic Geometry and its Applications*, John Wiley & Sons, 1996.

[6] W. Abbas, and M. Egerstedt, “Distribution of agents in heterogeneous multiagent systems,” In *Proc. IEEE Conf. Decision and Control*, pp. 976–981, Orlando, FL., 2011.

[7] P. Salvìk, “A tight analysis of the greedy algorithm for set cover,” *Journal of Algorithms*, vol. 25, no. 2, pp. 237–254, 1997.

[8] F. Kuhn, *The Price of Locality: Exploring the Complexity of Distributed Coordination Primitives*, PhD thesis, ETH Zürich, 2005.

[9] B. Liang and Z. J. Haas, “Virtual backbone generation and maintenance in ad hoc network mobility management,” In *Proc. IEEE Intl. Conf. Computer Communications*, Tel Aviv, Israel, 2000.

[10] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[11] L. Jia, R. Rajaraman, and T. Suel, “An efficient distributed algorithm for constructing small dominating sets,” *Distributed Computing*, vol. 15, no. 4, pp. 193–205, 2002.

[12] F. Kuhn and R. Wattenhofer, “Constant-time distributed dominating set approximation,” *Distributed Computing*, vol. 17, pp. 303–310, 2003.