Handling Mobility in Wireless Sensor and Actor Networks

Tommaso Melodia, *Member*, *IEEE*, Dario Pompili, *Member*, *IEEE*, and Ian F. Akyildiz, *Fellow*, *IEEE*

Abstract—In Wireless Sensor and Actor Networks (WSANs), the collaborative operation of sensors enables the *distributed sensing* of a physical phenomenon, while actors collect and process sensor data and perform appropriate actions. WSANs can be thought of as a distributed control system that needs to timely react to sensor information with an effective action. In this paper, coordination and communication problems in WSANs with mobile actors are studied. First, a new location management scheme is proposed to handle the mobility of actors with minimal energy expenditure for the sensors, based on a hybrid strategy that includes *location updating* and *location prediction*. Actors broadcast location updates limiting their scope based on Voronoi diagrams, while sensors predict the movement of actors based on Kalman filtering of previously received updates. The location management scheme enables efficient geographical routing, and based on this, an optimal energy-aware forwarding rule is derived for sensor-actor communication. Consequently, algorithms are proposed that allow controlling the delay of the data-delivery process based on power control, and deal with network congestion by forcing multiple actors to be recipients for traffic generated in the event area. Finally, a model is proposed to optimally assign tasks to actors and control their motion in a coordinated way to accomplish the tasks based on the characteristics of the events. Performance evaluation shows the effectiveness of the proposed solution.

Index Terms—Wireless sensor and actor networks, mobility, energy efficiency, real-time communications.

1 Introduction

Wireless Sensor and Actor Networks (WSANs) [2] are distributed wireless systems of heterogeneous devices referred to as *sensors* and *actors*. Sensors are low-cost, low-power, multifunctional devices that communicate untethered in short distances. Actors collect and process sensor data and consequently perform actions on the environment. In most applications, actors are resource-rich devices equipped with high processing capabilities, high transmission power, and long battery life.

In WSANs, the collaborative operation of the sensors enables the *distributed sensing* of a physical phenomenon. After sensors detect an event that is occurring in the environment, the event data are distributively processed and transmitted to the actors, which gather, process, and eventually reconstruct the event data. The process of establishing data paths between sensors and actors is referred to as *sensor-actor coordination* [2]. Once the event has been detected, the actors coordinate to reconstruct it, to estimate its characteristics, and make a collaborative decision on how to perform the action. This process is referred to as *actor-actor coordination* [2]. As a result, the operation of a WSAN can be thought of as an event-sensing, communication, decision, and acting loop.

Several applications for WSANs are concerned with *enhancing and complementing existing sensor network applications*. In these applications, the performed actions serve the purpose of enhancing the operation of the sensor network by enabling or extending its monitoring capability. For example, mobile actors can accurately deploy sensors [3], enable adaptive sampling of the environment [4], pick up data from the sensors when in close range, buffer it, and drop off the data to wired access points [5], or perform energy harvesting [6], or enhance the localization capabilities of sensors [7].

Conversely, we are concerned with new applications where actors are part of the network and perform actions based on the information gathered by sensors. We envision that WSANs will be an integral part of systems such as battlefield surveillance, nuclear, biological or chemical attack detection, home automation, and environmental monitoring [2]. For example, in fire detection applications, sensors can relay the exact origin and intensity of the fire to water sprinkler actors that will extinguish the fire before it spreads. Moreover, sensors can detect plumes, i.e., visible or measurable discharges of contaminants in water or in the air, and actors can reactively take countermeasures. Similarly, motion, acoustic, or light sensors in a building can detect the presence of intruders and command cameras or other instrumentations to track them. Alternatively, mobile actors can be moved to the area where the intruder has been detected to get high-resolution images, prompt or block the intruder. As a last example, in earthquake scenarios, sensors can help locate survivors and guide mobile actors performing rescue operations.

As an abstraction of several application setups encountered in the above-mentioned applications, we refer to a scenario where sensors monitor a given terrain and send samples of the event to the actors deployed on the terrain whenever an event occurs. Actors distributively reconstruct

- *T. Melodia is with the Department of Electrical Engineering, University at Buffalo, State University of New York, 332 Bonner, Buffalo, NY 14260. E-mail: tmelodia@eng.buffalo.edu.*
- *D. Pompili is with the Department of Electrical and Computer Engineering, Rutgers, State University of New Jersey, 94 Brett Road, Piscataway, NJ 08854. E-mail: pompili@ece.rutgers.edu.*
- *I.F. Akyildiz is with the Broadband Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. E-mail: ian@ece.gatech.edu.*

Manuscript received 3 Mar. 2008; revised 24 Nov. 2008; accepted 27 Apr. 2009; published online 27 May 2009.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-03-0071.
Digital Object Identifier no. 10.1109/TMC.2009.102.

the event based on partial information available at different actors, estimate the event characteristics, and identify an *action area*. Based on this, actors collaboratively decide on which actors should move to the action area and at which speed. The coordinated mobility of actors is thus triggered by the occurrence of events. Actors keep receiving event data until the event is active, and multiple consecutive events trigger subsequent reassignment of tasks among the actors.

In our prior work on WSANs [8], we introduced a framework for communication and coordination problems with static WSANs. The concepts of *sensor-actor coordination* and *actor-actor coordination* were introduced, and centralized optimal solutions and distributed heuristics were proposed. However, many challenging applications require support for mobile actors, which is not provided in [8]. Hence, in this paper, we extend our previous work in several directions.

First, we introduce a hybrid location management scheme to handle the mobility of actors with minimal energy consumption for the sensors. The proposed solution is tailored for WSAN applications and overcomes the drawbacks of previously proposed localization services [9], [10]. Actors broadcast updates limiting their scope based on Voronoi diagrams, while sensors predict the movements of actors based on Kalman filtering of previously received updates. Our proposed scheme combines joint use of Kalman filtering with Voronoi scoping on sensors and actors to lead to a new location management technique, which is shown to consistently reduce the energy consumption on sensors by avoiding over 75 percent of location updates with respect to existing location update algorithms.

The location management scheme is designed to enable efficient geographical routing for sensor-actor communications. Based on this, the second contribution of this paper is the development of an integrated routing/physical layer scheme for sensor-actor communication based on geographical routing, which is suited for mobile WSANs, and which leverages the information provided by the location management scheme. We derive a simple yet optimal forwarding rule based on geographic position in presence of Rayleigh fading channels. With respect to the previously proposed geographic forwarding rules [11], [12], our rule is optimal from the energy consumption standpoint. Furthermore, we show how to control the delay of the data-delivery process based on power control, i.e., to trade optimal energy consumption for decreased delay in case of low or moderate traffic. In case of high traffic, we introduce a new network congestion control mechanism at the network layer that forces multiple actors to share the traffic generated in the event area. This is shown to reduce delay, packet drops, and energy consumption even when traffic is sent to actors that are suboptimal from a network layer standpoint.

As a last contribution in our proposed system architecture, a new model for actor-actor coordination is introduced that enables coordinating motion and action of the participating actors based on the characteristics of multiple, concurrent events. In particular, the proposed model selects the best actor team to perform the required actions, based on the characteristics of the event, while trying to select the team of actors that will cause minimal reconfiguration of network operations. It drives the motion of the team toward the relevant area.

The paper is organized as follows: In Section 2, we review related work. In Section 3, we describe the proposed location management scheme, while in Section 4, we describe the sensor-actor communication solution. In Section 5, we introduce the actor-actor coordination model. In Section 6, we present performance evaluation results, while in Section 7, we conclude the paper.

2 Related Work

As discussed in [13], there are many open research challenges to enable real-time communication and coordination in sensor networks, especially due to resource constraints and scalability issues. Although a few recent papers are specifically concerned with coordination and communication problems in sensor and actor networks, the literature on the subject is very limited. In [2], research challenges in wireless sensor and actor networks are outlined and open research issues are described. In particular, several application scenarios are outlined, along with challenges for effective sensor-actor coordination and actor-actor coordination.

In [14], the authors deal with the problem of "hazards," which consist of out-of-order execution of queries and commands due to the lack of coordination between sensors and actors. In [15], the problem of mutual exclusion in WSANs is considered, which consists of determining the minimum subset of actors that covers the entire event region such that there is minimal overlap in the acting regions. An example would be poison gas actors, where one dose of the gas merely invalidates the subject, but two doses can kill. However, the proposed model does not consider mobile actors. A delay-energy aware routing protocol (DEAP) designed for sensor and actor networks is presented in [16], which enables a wide range of trade-offs between delay and energy consumption, including an adaptive energy management scheme that controls the wake-up cycle of sensors based on the experienced packet delay. However, the paper only focuses on sensor-actor communication and assumes predetermined sensor-actor associations.

Some recent papers [17], [18] have considered the issue of real-time communication in sensor networks. The SPEED protocol [17] provides real-time communication services and is designed to be a stateless, localized algorithm with low control overhead. End-to-end soft real-time communication is achieved by maintaining a desired delivery speed across the sensor network through a combination of feedback control and nondeterministic geographic forwarding. MMSPEED [18] is an extension of SPEED that can differentiate between flows with different delay and reliability requirements. SPEED and MMSPEED try to provide real-time delivery of individual flows from different sensors. Conversely, our solution is based on a collective notion of reliability that is associated with the overall event and not with each individual flow. Besides, none of these papers deals with sensor-actor coordination, i.e., defines how actors and sensors coordinate and communicate, or with actor-actor coordination.

3 Location Management

The network is composed of N_S sensors and N_A actors, with $N_S >> N_A$. Each sensor is equipped with a low data rate radio interface. Actors are equipped with two radio transmitters, i.e., a low data rate transmitter to communicate with the sensors and a high data rate wireless interface for actor-actor communication. From the perspective of sensors, actors are *equivalent recipients of information*. Hence, each sensor will route information to its closest actor, unless an alternative actor is preferable in case of congestion, as described later.

In line with recent work on routing algorithms for sensor networks [8], [11], [12], [19], we study the *sensor-actor coordination* based on a geographical routing paradigm. Geographical routing algorithms are attractive especially for their *scalability* since routing decisions are inherently *localized* [19]. The scalability of geographical routing protocols is apparent in static sensor networks with a single sink. In networks with mobile nodes and multiple recipients, however, it depends on the ability of location management schemes to efficiently provide relevant nodes with the position of mobile nodes at any time. Previous proposals have dealt with the development of scalable location services for tracking mobile nodes in distributed systems based on geographical routing. In [9], GLS was proposed, which is a hierarchical location service where each mobile node maintains its current location in a number of location servers distributed throughout the network. In [10], the performance of GLS is compared to two other location services based on similar premises. In general, the objective of these mechanisms is to potentially allow each single device in the network to retrieve the location of any other node. We argue that the extensive message exchange and complex server structure, often hierarchical, associated with these protocols, can be avoided given the characteristics of WSANs.

In general, location management may follow two strategies: *location updating* and *location prediction*. Location updating is a passive strategy in which each actor periodically broadcasts its position to the neighboring sensors. Location prediction is a dynamic strategy in which sensors proactively estimate the location of their neighboring actors. In this case, the tracking efficiency depends on the accuracy of the mobility model and on the efficiency of the prediction algorithm. Our proposed solution is based on a hybrid scheme. The underlying principle is to leverage the characteristics of WSANs to minimize location updates in the spatial and temporal domains, since every location update causes energy consumption at the receiving sensors, and may lead to the *broadcast storm* problem when update messages need to be relayed throughout the network. For this reason, we propose a proactive location management approach based on update messages sent by mobile actors to sensors. As discussed, in WSANs, each actor is an equivalent recipient of information. Therefore, sensor-actor communications are localized, i.e., each sensor sends information to its closest actor. Hence, in the spatial domain, broadcasts can be limited based on Voronoi diagrams [20]. At the same time, actor movement is to some extent predictable, as it is driven by the actor-actor coordination procedures. Hence, in the temporal domain, location updates can be limited to *actor positions that cannot be predicted* at the sensor side. Location updates are triggered at the actors when the actual position of the actor is "far" from what can be predicted at the sensors based on past measurements. Therefore, actors that move following predictable trajectories, which is likely to be a common case in WSANs, as will become clearer in Section 5 will need to update their position much less frequently than actors that follow temporally uncorrelated trajectories.

3.1 Limiting Broadcasts in Space

We use Voronoi diagrams to limit the scope of actor-initiated location updates. The Voronoi diagram of a set of discrete sites partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one site. For their properties and ease of computation, Voronoi diagrams have been previously applied to the area of sensor networks. For example, in [21], they are used along with Delaunay triangulation to study sensor network coverage. In [22], Voronoi diagrams are used in connection with the concept of exposure, i.e., a measure of how well an object, moving on an arbitrary path, can be observed by the sensor network over a period of time. In [23], an optimal polynomial-time worst- and average-case algorithm for coverage calculation with homogeneous isotropic sensors is derived. Moreover, Voronoi Diagrams and Delaunay triangulation have been used in geographical routing to obtain subgraphs with desirable properties [24]. Instead, we leverage Voronoi diagrams to limit the spatial extension of actor broadcasts.

The Voronoi cell of an actor a_i contains all points of the plane that are closer to a_i than to any other actor in the network. A sensor s is said to be *dominated* by an actor a if its location lies in the Voronoi cell of a. Every actor is responsible for location updates to sensors in its Voronoi cell and regulates its power so as to limit interference beyond the farthest point in its Voronoi cell. Each sensor will thus expect to receive location updates from the actor it is dominated by. With respect to flooding, the energy consumption for location updates is drastically reduced. With a flooding-like protocol, each actor sends a message to its N neighboring sensors. We consider the link metric $E = 2E_{\mathrm{elec}} + E_{\mathrm{amp}}d^{\alpha}$, where α is the path loss propagation exponent ($2 \leq \alpha \leq 5$), E_{amp} is a constant $[J/(bits \cdot m^{\alpha})]$, and E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit $[J/bits]$. Each sensor, upon receiving the message, forwards it by broadcasting again. On this first hop only, the energy consumption is $N_A \cdot (NE_{elec} + N(E_{elec} + E_{amp}d^{\alpha} + NE_{elec}) = N_A \cdot (2NE_{elec} + NE_{amp}d^{\alpha} + N^2E_{elec})$. At least we need a message from each actor to reach each sensor in the network, and the same message can potentially be relayed to each other node in the network before it is discarded. This is clearly a worst-case scenario, but it provides an indication of the scaling law for the energy consumption. Instead, provided that each actor can transmit data within its Voronoi cell, no forwarding is needed, and hence, the energy consumption is in the order of the number of sensors (energy needed to receive the update packets). Hence, the worst-case energy consumption of a flooding scheme increases as a function order of $O(N_S^2 \cdot N_A)$, and most of the energy burden is on the sensors. Conversely, if the actor is able to reach all sensors in its Voronoi cell in one hop, which may be true in many practical cases, the energy consumption increases as a function order of $O(N_S)$, and most of the energy burden is on the actors.

3.2 Limiting Broadcasts in Time

In the temporal domain, location updates can be limited to *actor positions that cannot be predicted* at the sensor side. Location updates can be triggered at the actors only when the actual position of the actor is "far" from what can be predicted at the sensors based on past measurements. Therefore, actors that move following predictable trajectories, which is likely to be a common case in WSANs, will need to update their position much less frequently than actors that follow temporally uncorrelated trajectories. In [25], adaptive and predictive protocols to control the frequency of localization based on sensor mobility behavior to reduce the energy requirements for localization while bounding the localization error are proposed. In addition, the authors evaluate the energy accuracy trade-offs that arise: intuitively, the higher the frequency of localization, the lower the error introduced because of mobility. Different from [25], we adaptively vary the frequency of location updates based on sensor-side Kalman filtering of previously received updates. Another interesting related work, originated in the database community, is presented in [26]. The authors propose a new abstraction called model-based views, which represents a model of the sensed phenomenon, and propose to report new reading only when the latter deviates from prediction inferred from the model. We further observe that Kalman filtering is used as a means for decentralized estimation of objects in sensor networks in [27], [28] and in wireless multimedia sensor networks in [29]. Note that while in these contributions Kalman filtering is used for object tracking, our work is concerned with the design of a localization mechanism to enable geographical routing in WSANs. Actors are assumed to be endowed with an onboard localization system (e.g., GPS), while sensors predict the position of actors based on Kalman filtering of sparse measurements (taken at the actor and transmitted to the sensors). As a last note, we would like to emphasize that our location management scheme can be applied even with prediction strategies different from the Kalman filter. For example, simpler linear filters such as auxiliary-vector filters [30] can be used when even lower computational complexity is desired, while extended Kalman filters can be designed in the presence of nonlinear measurement or movement models.

The dynamic movement model for the ith actor in two-dimensional coordinates can be described by a continuous-time linear dynamical system. The equivalent discrete-time dynamic equation can be derived as in [31] by means of the state space method. Hence,

$$\mathbf{x_i^k} = \mathbf{F}\mathbf{x_i^{k-1}} + \mathbf{G}\mathbf{u_i^{k-1}} + \mathbf{B}\mathbf{w_i^{k-1}} \tag{1}$$

represents the state transition equation for the system describing the motion of actor i between steps $k-1$ and k, where

$$\mathbf{F} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}, \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

In (1), $\mathbf{x_i^k} = [x_i^k, y_i^k, \dot{x}_i^k, \dot{y}_i^k]^T$ represents position and velocity of actor i at step k; $\mathbf{u_i^k} = [u_i^{k,x}, u_i^{k,y}]^T$ represents the control input for $t \in [kT, (k+1)T)$, where T is the sampling interval; and $\mathbf{w_i^k} = [w_i^{k,x}, w_i^{k,y}]^T$ represents discrete random acceleration caused by environmental noise or nonidealities in the control input. The variable $\mathbf{w_i^k}$ represents two-dimensional samples of discrete-time white Gaussian noise. Hence, $\mathbf{w_i^k} \sim \mathcal{N}(0, \mathbf{Q})$, with $\mathbf{Q} \geq 0$, where $\mathbf{Q}$ is the covariance matrix of the process. The random acceleration is also assumed to be independent on the two axes.

The position observed by the actor at step k is related to the state by the *measurement equation*

$$\mathbf{z_i^k} = \mathbf{H}\mathbf{x_i^k} + \mathbf{C}\mathbf{v_i^k}, \tag{3}$$

where $\mathbf{z_i^k} = [z_i^{k,x}, z_i^{k,y}]$ represents the *observed position* of the actor at step k, and where $\mathbf{H} = [\mathbf{I} \quad 0],\ \mathbf{C} = \mathbf{B}$.

The variable $\mathbf{v_i^k} = [v_i^{k,x}, v_i^{k,y}]^T$ represents the *measurement noise*, expressed as two-dimensional samples of discrete-time white Gaussian noise. Hence, $\mathbf{v_i^k} \sim \mathcal{N}(0, \mathbf{R})$, with $\mathbf{R} \geq 0$, where $\mathbf{R}$ is the covariance matrix of the process. The observed position of the actor $\mathbf{z_i^k}$ is, thus, the actual position of the actor affected by a measurement noise, which we represent as a Gaussian variable. Note that to keep the model general, we do not assume a particular localization technique for the actor, e.g., GPS, particle filtering [32], among others.

The Kalman filter provides a computationally efficient set of recursive equations to estimate the state of such process, and can be proved to be the optimal filter in the minimum square sense [33]. The joint use of Kalman filter at the sensor and actor sides enables reducing the number of necessary location updates. In fact, the filter is used to *estimate the position* at the actor based on measurements, which is a common practice in robotics, and to *predict* the position of the actors at the sensors, thus, reducing the message exchange. The position of actor i can be estimated and predicted at the sensors in its Voronoi cell, based on the measurements $\mathbf{z_i^k}$ taken at the actor and broadcast by the actor. At step k, each sensor s in i's Voronoi cell updates the state (that represents position and velocity of the actor) based on the equations

$$\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-}} = \mathbf{F}\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-1}}, \qquad \mathbf{P}_{\mathbf{i,s}}^{\mathbf{k-}} = \mathbf{F}\mathbf{P}_{\mathbf{i,s}}^{\mathbf{k-1}}\mathbf{F^T} + \mathbf{Q}. \tag{4}$$

Equation (4) describes how sensor s predicts the state of actor i before receiving the measurement (a priori *estimate*). Note that the control input $\mathbf{u_i^{k-1}}$ is not known at the sensor, while it is used at the actor to update the state. Then, sensor s projects the covariance matrix ahead. After receiving the measurement from actor $\mathbf{z_i^k}$, sensor s updates the Kalman gain $\mathbf{K_{i,s}^k}$ and corrects the state estimate and covariance matrix according to the measurement, i.e.,

$$\mathbf{K_{i,s}^k} = \mathbf{P_{i,s}^{k-}}\mathbf{H^T}\left(\mathbf{H}\mathbf{P_{i,s}^{k-}}\mathbf{H^T} + \mathbf{R}\right)^{-1}, \tag{5}$$

$$\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-}} + \mathbf{K_{i,s}^k}\left(\mathbf{z_{i,s}^k} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-}}\right), \tag{6}$$

$$\mathbf{P_{i,s}^k} = \left(\mathbf{I} - \mathbf{K_{i,s}^k}\mathbf{H}\right)\mathbf{P_{i,s}^{k-}}. \tag{7}$$

In particular, (5) updates the Kalman gain, (6) calculates the new state (a posteriori *estimate*), while (7) updates the covariance matrix. Note that the complexity of the above computations is very low as the number of state variables is only 4. Moreover, the processing cost for sensors is much lower than the communication cost. This is justified by Pottie and Kaiser [34], where the energy necessary to transmit 1 Kbit on $100\ \mathrm{m}$ is shown to be equivalent to the energy necessary to execute 300,000 processor instructions on an 8-bit processor such as those used by MICAz motes [35].

At each step k, each actor i emulates the prediction procedure performed at the sensors in its cell, calculates its actual new position by filtering the new measurement, and broadcasts the new measurement $\mathbf{z_i^k}$ if and only if a sensor s in its cell, which has received the previous updates, is not able to predict the position of the actor within a maximum error e_{max}, i.e., if $(\mathbf{z_i^k} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-}}) > e_{max}$. If sensor s does not receive a location update at step k, it assumes that $\mathbf{z_i^k} = \mathbf{H}\hat{\mathbf{x}}_{\mathbf{(i,s)}}^{\mathbf{k-}}$, i.e., the predicted position coincides with the actual new position of the actor. Based on this, it updates its estimate of the state for actor i as in (5)-(7).

4 Sensor-Actor Communication

The location management scheme discussed in Section 3 is designed to enable efficient geographical routing for sensor-actor communications, techniques that are particularly well-suited for mobile WSANs. By leveraging localization information provided by this location management scheme, we concentrate on studying integrated routing/physical layer schemes for sensor-actor communication based on geographical routing. Based on this, we derive a simple yet optimal forwarding rule based on geographic position in presence of Rayleigh fading channels. When the timeliness of received information is an issue, we propose an algorithm to reduce delay by means power control at low loads, while spatial diversity of different actors is used to reduce delay/congestion at higher loads when power control is not sufficient. To the best of our knowledge, this idea has not been considered before.

In [8], we proposed a new notion of reliability that accounts for the percentage of packets generated by the sensors in the event area that are received within a predefined latency bound. The *event reliability* r perceived by an actor is the ratio of *reliable* data packets over all the packets received in a decision interval, where a packet is considered reliable if it is received within a given latency bound. The *event reliability threshold* r_{th} is the minimum event reliability required by the application. Unlike other more conventional notions of reliability, this definition is related to the timely delivery of data packets from sources to actors, and is calculated at the network layer. Note that we do not aim at devising a solution that guarantees full reliability or that provides hard real-time guarantees on data delivery. Rather, the objective is to trade off energy consumption for latency when data have to be delivered within a given time bound B with a given reliability r_{th}. The solution presented in [8], based on similar premises, is however not suitable for mobile actors, as the convergence of the distributed protocol to an energy-efficient and latency compliant solution is too slow as compared to the dynamics encountered in networks with mobile actors. Therefore, when the traffic generated in the event area is low or moderate, we adjust the end-to-end delay by increasing the forwarding range with respect to the energy-efficient forwarding range, as described in Section 4.1. We propose an algorithm to accomplish this that is based on collective feedback from the corresponding actor. Then, in Section 4.2, in case of congestion at a recipient actor, we reduce the end-to-end delay by rerouting part of the traffic to another, less congested, actor.

4.1 Power-Controlled Energy-Delay Adjustment

Previous work on geographical routing considered primarily greedy forwarding[1] whereby a packet is forwarded to the closest node to the destination. However, this usually entails selecting links that connect the forwarding node to neighbors that reside close to the border of the connectivity range. When a realistic model of the effects of wireless propagation is considered, such links are likely to be unstable and prone to high packet error rates. Hence, the authors of [11], [12] propose enhanced flavors of greedy forwarding that avoid using those links. However, the objective is still to maximize the advance toward the destination, while we propose to forward packets on energy-efficient links, by trading off advancement at every single hop to minimize the energy consumption, unless a higher advancement is needed to increase the reliability. Moreover, as in [11], [12], [37], we remove the *unit disk graph* assumption relied on by most routing researches and consider a more accurate connectivity model. Local metrics for energy-efficient geographic forwarding are derived in [38]. However, the authors of [38] focus on networks with relatively stable wireless channels, which is a practical assumption when a wireless network is in an isolated remote environment with either slow moving or no mobility events. Conversely, we derive the energy-efficient forwarding distance in the presence of a fast fading channel. In addition, we propose a mechanism to decrease the end-to-end delay by increasing the transmit power. Finally, we note that our work is related to [39], where a heuristic is developed for an anycast base station selection optimization problem. In our scheme, however, the geographically closest actor is used as a base station, and traffic is partially rerouted to an alternative base station in case of congestion as will be discussed in Section 4.2.

Let us refer to the communication between v_i (forwarder) and v_j. If we denote their distance by d_{ij}, the probability $\mathcal{P}_{ij}^s$ that node v_j will receive a packet transmitted by v_i can be expressed as

$$\mathcal{P}_{ij}^s = \Pr\left\{ \frac{P_{ij}^t \cdot f}{\beta d_{ij}^{\alpha}} \geq \Gamma \right\}, \tag{8}$$

where $P_{ij}^t\ [\mathrm{W}]$ is the power transmitted at $v_i, \Gamma\ [\mathrm{W}]$ is a technology-dependent parameter representing the receiver threshold, βd_{ij}^{α} represents the path loss, with $\beta[\mathrm{m}^{-\alpha}]$ representing a dimensional parameter, while f is a unit-mean Rayleigh distributed r.v. that models fast fading for a given packet. We assume the so-called *block fading model*, i.e., the attenuation due to fading remains constant during a

1. Greedy forwarding has been enhanced in [36] by introducing face/perimeter routing techniques to route packets around the void area to reach the destination. This technique can be applied to the mechanism proposed in this paper in low-density or concave areas.

packet transmission, but it is uncorrelated among subsequent transmission events. Hence, we can write

$$\mathcal{P}^s_{ij} = \Pr\left\{ f \geq \frac{\Gamma\beta d^\alpha_{ij}}{P^t_{ij}} \right\} = \int_{\frac{\Gamma\beta d^\alpha_{ij}}{P^t_{ij}}}^{+\infty} p_f(f)df = e^{-\frac{\pi}{4}\left(\frac{\Gamma\beta d^\alpha_{ij}}{P^t_{ij}}\right)^2}. \quad (9)$$

The transmit power P^t_{ij} is related to the distance-dependent energy consumption through the transmit rate R as $P^t_{ij} = E_{\text{amp}} \cdot d^\alpha_{ij} \cdot R$. We can interpret $E_{\text{amp}} \cdot d^\alpha_{ij} \cdot R$ as the power necessary to transmit a bit over a distance d_{ij}, given a target packet error rate. The expression can be generalized by including a term that allows adjusting the desired bit error rate as follows:

$$P^t_{ij} = (E_{\text{marg}} + E_{\text{amp}}) \cdot d^\alpha_{ij} \cdot R. \quad (10)$$

A higher value for E_{marg} leads to a higher energy consumption, and at the same time, increases the probability of successful reception at the receiver, thus, decreasing the expected number of retransmissions. By substituting (9) into (10), we obtain

$$\mathcal{N}^{RTX}_{ij}(d, E_{\text{marg}}) = \frac{1}{1 - PER_{ij}} = \frac{1}{\mathcal{P}^s_{ij}} = e^{\frac{\pi}{4}\left[\frac{\Gamma\beta}{(E_{\text{marg}}+E_{\text{amp}})\cdot R}\right]^2}, \quad (11)$$

where PER_{ij} denotes the packet error rate on link ij. Now, consider a node v_i forwarding a packet toward a destination actor a_k at distance D. The latter is available at each sensor node through the location update mechanism described in Section 3. The end-to-end energy consumption can then be expressed as

$$E_{\text{e-e}} = \sum_{(i,j)\in\mathcal{P}(v_i,a_k)} \left(\frac{P^t_{ij}}{R} + 2E_{\text{elec}} \right), \quad (12)$$

where $\mathcal{P}(v_i, a_k)$ represents the path between v_i and a_k. Ideally, the end-to-end energy consumption is minimized when data are forwarded on a set of nodes located on the line connecting the source and the destination, equally spaced with internode distance d^{opt}. By substituting (10) into (12), and by considering retransmissions, we obtain

$$E^{\min}_{\text{e-e}} = \min_{d,E_{\text{marg}}} \left\{ \frac{D}{d_{ij}} \left[2E_{\text{elec}} + (E_{\text{marg}} + E_{\text{amp}}) d^\alpha_{ij} \right] \cdot \mathcal{N}^{RTX}_{ij} \right\},$$

where $\mathcal{N}^{RTX}_{ij}$ is given by (11). The values $(d^{opt}, E^{opt}_{\text{marg}})$ that minimize the above expression can be found by solving the nonlinear system $\nabla E_{\text{e-e}} = \mathbf{0}$, i.e., $[\frac{\partial E_{\text{e-e}}}{\partial d}, \frac{\partial E_{\text{e-e}}}{\partial E_{\text{marg}}}] = [0, 0]$, to find the stationary points of the function. A sufficient condition for a stationary point to be a minimum is that $\nabla^2 E_{\text{e-e}} \succ 0$, i.e., the Hessian calculated at the stationary point is positive definite. Note that the *optimal forwarding distance* d^{opt} is independent of D, i.e., the distance between the forwarding node and the intended destination. The expression can be interpreted as the optimal trade-off between distance-independent and distance-dependent energy consumptions, and lends itself well to the development of localized forwarding rules. In case of ideal channel, and with $E_{\text{marg}} = 0$, (13) is minimized when

$$d^{\text{opt}} = \sqrt[\alpha]{\frac{2 \cdot E_{\text{elec}}}{E_{\text{amp}}(\alpha - 1)}}.$$

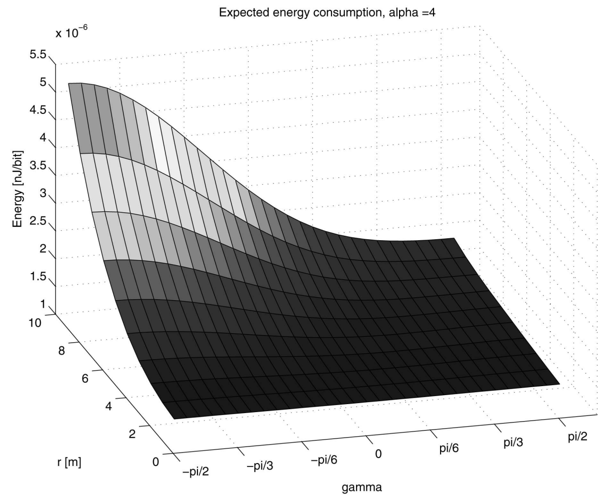

Fig. 1. Energy consumption with varying angle and distance from optimal forwarding point.

With the parameters given in [40], i.e., $E_{\text{elec}} = 50$ nJ/bit, $E_{\text{amp}} = 100$ pJ/bit/m$^\alpha$, $\alpha = 2.5$, the optimal forwarding distance for an ideal channel is $d^{\text{opt}} = 13.47$ m. Solving (13) yields $d^{\text{opt}} = 8.00$ m and $E^{\text{opt}}_{\text{marg}} = 86$ pJ/bit/m$^\alpha$, i.e., $E^{\text{opt}}_{\text{marg}} + E_{\text{amp}} \approx 2E_{\text{amp}}$. Hence, as expected, the optimal forwarding distance on a Rayleigh fading channel is lower than with an ideal channel, and a higher transmission power is needed. It can be concluded that the energy-optimal path is obtained by forwarding the packet to a node that is located d^{opt} meters away on the line connecting the forwarding node and the destination. We refer to this point on the 2D plane as the *optimal forwarding point*. A practical forwarding rule should intuitively select the next hop with minimal distance from this point. However, Fig. 1 shows, when $\alpha = 4$, the expected end-to-end energy consumption with varying position of the next hop with respect to the optimal forwarding point. This is expressed in terms of the distance r from the optimal forwarding point and of the angle γ formed between the line connecting source and destination and the line connecting the next hop to the optimal point. An angle $\gamma = -\pi/2$ indicates a next hop on the line connecting the source and destination but farther from the source than the optimal point, while $\gamma = \pi/2$ indicates a next hop on the line connecting the source and destination but closer than the source to the optimal point. As shown in Fig. 1, when α is high, it is important to avoid nodes that are farther from the source than the optimal point. Conversely, when α is lower than 3.5, the closest node to the optimal forwarding point is also energy optimal. In the following, we propose an algorithm to find an energy-latency trade-off, which relies on end-to-end feedback from the actors advertising their reliability:

Algorithm 1. Optimal forwarding for node v_i

Given:

v_i, the set of neighbors of v_i $\mathcal{N}(v_i)$, and the set of actors $\mathcal{A}$:

$k^* = argmin_k(\delta(v_i, a_k)), a_k \in \mathcal{A}$

$\alpha = \tan^{-1}\frac{(y_{k^*} - y_i)}{(x_{k^*} - x_i)}$

$x^{\text{opt}} = x_i + d^{\text{opt}} \cdot \cos\alpha$

$y^{\text{opt}} = y_i + d^{\text{opt}} \cdot \sin\alpha$

$j^* = argmin_j(\delta([x^{\text{opt}}, y^{\text{opt}}], v_j)), v_j \in \mathcal{N}(v_i) \cap \mathcal{P}(v_i, a_k)$

4.1.1 Feedback-Controlled Energy-Delay Adjustment

According to Algorithm 1, each sensor node v_i selects its closest actor a_k^* as its destination (where $\delta()$ indicates euclidean distance). Then, it calculates the angle α formed by the ideal line connecting itself and the destination actor, and a reference direction. It then calculates the optimal forwarding point by projecting d^{opt} in the direction of a_k^*. The optimal forwarding point $\mathbf{x}^{\mathrm{opt}}$ in the figure is at distance d^{opt} from v_i on the line toward a_k^*. Finally, the next hop v_{j*} is selected as the closest neighbor with positive advance to the optimal forwarding point. Note that $\mathcal{P}(v_i, a_k)$ represents the set of nodes with positive advance toward a_k with respect to v_i.

Algorithm 2 describes how to control the reliability by means of actor feedback messages. We adopt a conservative approach. When an event occurs, all sensors start transmitting with the maximum forwarding range. Then, according to the actor feedback on the observed reliability, sensors may decrease their forwarding range until either the reliability is close to the required event reliability threshold r_{th} or the optimal forwarding range is reached. Transmitting closer than the optimal forwarding range, as will be shown in Section 6.1, leads to high delay and high energy consumption, and is thus avoided. When the observed reliability is low even with the longest forwarding ranges, the actor initiates procedures for network layer congestion control, as explained in Section 4.2.

Algorithm 2. Reliability control
  $d = d^{max}$
  Calculate reliability r_i
  while $(r_i > r_{th} - \epsilon)$ and $(d > d^{opt})$ **do**
    $d = d - \Delta d$
  end while
  while $r_i \leq r_{th}$ **do**
    Calculate optimal actor a_{k*}
    Send *virtual position* $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$
  end while

4.2 Network Layer Actor-Driven Congestion Control

In several application scenarios, high sampling rates at the sensors, large event areas, or dense deployment may lead to high contention and consequent collisions at the MAC layer, and ultimately, to decreased reliability. In classical network theory, these situations are usually handled by decreasing the data rate by means of congestion control algorithms at the transport layer. However, although congestion control mechanisms have been devised for sensor networks [41], these usually rely on spatial correlation among sampled data and assume that the sampling rate at the sensors can be changed. Nevertheless, the peculiar characteristics of WSANs, and in particular, the equivalence of different actors as recipients for sensor data, allow devising procedures to relieve congested actors from excessive traffic burden by deviating traffic toward other idle actors. Indeed, the objective of such a procedure is to trade off energy consumption, by reaching a suboptimal actor, for increased reliability. To do so, there is a need to develop a mechanism to allow congested actors to detect situations of congestion, to identify suitable alternate actors to reroute traffic to, and to notify sensors that a different actor needs to receive their data. In this section, we propose a mechanism to take countermeasures at the network layer.

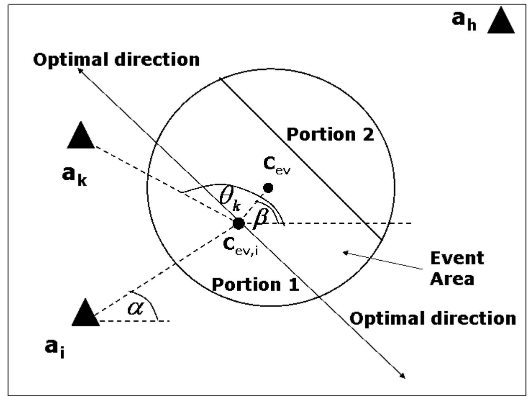

Fig. 2. Calculation of directivity factor δ_i.

We propose to detect congestion at the actor receiving data and redirecting traffic to other, less congested actors. We consider the notion of reliability from [8], as recalled at the beginning of this section. Whenever an actor a_i detects very low reliability, caused by excessive delays and packet drops, it selects another actor to reroute the traffic from half of the sensors in its Voronoi cell to that actor. Each actor a_k is assigned by a_i a weight w_k, which measures its suitability to become a recipient for the traffic generated in the portion of the event area that a_i is receiving data from. The weight w_k, which is low for better suited actors, is calculated as the weighted sum of three factors, $w_k = \frac{c_\eta \eta_k + c_\delta \delta_k + c_\Delta \Delta_k}{c_\eta + c_\delta + c_\Delta}$, with weights c_η, c_δ, and c_Δ. As a design choice, we set $c_\eta \geq c_\delta \geq c_\Delta$.

1. *Congestion factor* $\eta_k, 0 \leq \eta_k \leq 1$. This normalized value reflects the reliability observed at actor a_k, i.e., $\eta_k = 1$ if $r < r_{th} - \epsilon$, it monotonically decreases as $r - r_{th}$ increases, and $\eta_k = 0$ for actors that are not receiving traffic. Here, ϵ represents a suitable reliability margin to prevent instability.
2. *Directivity factor* δ_k, which reflects the relative angular position of actor a_k with respect to actor a_i and the center of the event area.

Let us refer to Fig. 2, which illustrates the situation where an actor a_i is receiving data from part of the event area. We indicate the center of the event area as C_{ev}, which represents the weighted sum of the positions of the sensors. The center of the portion of the event area that resides in a_i's cell is referred to as $C_{ev,i}$. In Fig. 2, the event area is divided into two parts, and another actor receives data from the second portion of the event area. However, the proposed procedure to calculate the directivity factor holds in the general case, where the event area is divided among multiple actors, given that the center of the global event C_{ev} has been collaboratively reconstructed by the participating actors. The idea is to give higher weights to actors that reside in the same direction of a_i with respect to $C_{ev,i}$, as this would cause increased traffic in the direction of a_i; or in the direction of C_{ev} with respect to $C_{ev,i}$, as this would increase traffic in the event area. Rather, the directivity factor should be maximum for those actors that are away from these two directions (optimal directions in Fig. 2). The angles α, β, and θ_k describe the relative angular positions of $C_{ev,i}$ and a_i, C_{ev}, and a_k, respectively. After some derivations, the directivity factor for actor a_k can be calculated as follows:

$$\delta_k = \begin{cases} \dfrac{2\theta_k + (\pi - \beta - \alpha)}{(\pi + \beta - \alpha)} & 0 \leq \theta_k \leq \beta, \\ \dfrac{|2\theta_k - (\pi + \beta + \alpha)|}{(\pi + \alpha - \beta)} & \beta \leq \theta_k \leq \pi + \alpha, \\ \dfrac{|2\theta_k - (3\pi + \alpha + \beta)|}{(\pi + \beta - \alpha)} & \pi + \alpha \leq \theta_k \leq 2\pi. \end{cases} \quad (13)$$

3. *Distance factor* Δ_k, which is the distance of the actor from the center of the event $C_{ev,i}$ normalized to the diameter of the monitored area, i.e., $\Delta_k = 1$ with maximum distance.

A congested actor a_i selects the optimal actor a_{k*} with minimum weight w_{k*}. Then, actor a_i calculates and advertises a new *virtual position* $\mathbf{x_{k*}^{virt}}$ for a_{k*} to the sensors in its Voronoi cell. The virtual position is forced to be on the line connecting the real position of the actor $\mathbf{x_{k*}}$ and the center of the event area $C_{ev,i}$, and corresponds to the point such that half of the sensors in $C_{ev,i}$ are closer to a_i, while the other half is closer to a_{k*}. Each sensor will select its recipient actor, using for actor a_{k*} the virtual position $\mathbf{x_{k*}^{virt}}$, while the real position $\mathbf{x_{k*}}$ is still used to perform the actual forwarding function. The concept of virtual position allows to optimally partition the sensors in such a way that only those that are closer to a_{k*} redirect their traffic to it, and provides a compact way to notify the sensors. The procedure is applied recursively by actors that are still congested after splitting the traffic into two.

Algorithm 3 describes the procedure run by actor a_i to calculate the virtual position for actor a_k. The symbols $\mathbf{x_i}$ and $\mathbf{x_{k*}}$ refer to the position of actors a_i and a_{k*}, while $\mathcal{S}_i$ refers to the set of sources that resides in the portion of the event area closer to a_i.

Algorithm 3. Calculate virtual position for actor a_{k*}

```
x_{k*}^{virt} = x_{k*}
x_k^{last} = x_{k*}
N_i = Calculate sensors in S_i closer to x_i
N_{k*}^{virt} = Calculate sensors in S_i closer to x_{k*}^{virt}
while |N_i - N_{k*}^{virt}| > 1 do
    if N_i > N_{k*}^{virt} then
        x_k^{last} = x_{k*}^{virt}
        x_{k*}^{virt} = (x_{k*}^{virt} + C_{ev,i})/2
    else
        x_{k*}^{virt} = (x_{k*}^{virt} + x_k^{last})/2
    end if
    N_i = Calculate sensors in S_i closer to x_i
    N_{k*}^{virt} = Calculate sensors in S_i closer to x_{k*}^{virt}
end while
```

5 Actor-Actor Coordination

As a last component of our system, in this section, we propose a model, based on MINLP, to coordinate actor mobility. Our coordination model assigns tasks to different actors, where a task represents 1) moving toward the event area identified by the sensor and 2) performing an action there (e.g., extinguish a fire) with certain required characteristics. We refer to the coordination problem as multiactor task allocation problem. The solution to this problem selects the best *actor team* that minimizes energy consumption while causing minimal reconfiguration to the current network operation, and to control their motion toward the action area. Our previous work [8] assumes that static actors are only able to act within a circular area defined by their action range. Hence, it is not suitable for WSANs with mobile actors. Moreover, in [8], reallocation of resources to face multiple events is not considered. Here, we introduce a more general framework and remove these assumptions.

The position of the sensors that generate readings defines the *event area*. The *action area* represents the area where the actors should act, and is identified by processing the event data. In general, the event and the action areas may be different, although they may coincide in several applications. We consider a scenario where multiple events may give rise to event/action areas partially overlapped in space and/or time, and an event may occur before the actions associated with previous events have been successfully completed. The proposed allocation problem presents analogies with the class of so-called *Multirobot Task Allocation* (MRTA) problems encountered in robotics [42]. We are concerned with methods for *intentional cooperation*, i.e., mobile actors cooperate explicitly through task-related communication and negotiation, and coordinate their motion to efficiently act on the action areas, based on the characteristics of the reconstructed events. Other approaches to cooperation, such as minimalist or emergent approaches [42], where individual actors coordinate their actions without explicit negotiation or allocation of tasks, are out of the scope of this paper.

According to the event features collected from the event area, each occurring event ω in the *event space* Ω can be characterized by the tuple $\mathcal{E}^{(\omega)} = \{F^{(\omega)}, Pr^{(\omega)}, A^{(\omega)}, S^{(\omega)}, I^{(\omega)}, D^{(\omega)}\}$, where $F^{(\omega)}$ describes the *event type*, i.e., the class the event belongs to, $Pr^{(\omega)}$ the *priority*, $A^{(\omega)}[m^2]$ the *event area*, $S^{(\omega)}[m^s]$ and $I^{(\omega)}[J/m^2]$ the *scope* (the action area) and *intensity*, respectively, and $D^{(\omega)}[s]$ the *action completion bound*, i.e., the maximum allowed time from the instant when the event is sensed to the instant when the associated action needs to be completed. These characteristics, which define each occurring event, are distributively reconstructed by the actors that receive sensor information, and constitute inputs to the multiactor task allocation problem. In particular, the multiactor allocation problem consists of selecting a *team of actors* and their *velocity* to optimally divide the action workload, so as to minimize the energy required to complete the action, while respecting the *action completion bound*. Although actors are resource-rich nodes, the order of magnitude of the energy required for actions and movements is higher than that required for communication. Hence, it is important to save action and movement energy to extend the lifetime of actors. We formulate the multiactor allocation problem as a *Mixed Integer Non-linear Program* (MINLP).

In the following, the objective is to find, for each occurring event $\omega \in \Omega$, the subset of actors and their optimal velocities in such a way as to minimize the energy required to complete the action associated with the occurring event, under the constraint of meeting the action completion bound. We rely on the following assumptions: 1) the energy to perform the action (action and movement energy) is orders of magnitude higher than the energy required for communication; 2) actors are able to *selectively* act on part of the action area they are assigned to; and 3) task reallocation is performed only if higher priority actions cannot be accomplished due to the lack of resources.

We introduce the following notations:

- l_a^f $[W]$ is the *action power level* of actor a when the event type $f \in \mathcal{F}^{(\omega)}$;
- $T_a^{\Omega,(\omega)}[s]$ is the time actor a needs to complete the action associated with event ω when a is part of an acting team;
- $E_a^{\Omega,(\omega)} = l_a^f \cdot T_a^{\Omega,(\omega)}[J]$ is the energy required by a to complete its task, given its action power level and action time;
- $d_a^{(\omega)}[m]$ is the distance between actor a and the center of the action area $S^{(\omega)}$, while $T_a^{M,(\omega)}[s]$ is the time needed by actor a to reach it;
- $E_a^{M,(\omega)} = [\beta v_a^{(\omega)\gamma} + P_{min}^M] \cdot T_a^{M,(\omega)}[J]$ is the energy actor a requires to move at speed $v_a^{(\omega)}$ for $T_a^{M,(\omega)}$ seconds, where $P_{min}^M[W]$ is a velocity-independent term that accounts for dissipative effects;
- $\mathbf{X}^{(\omega)}$ is a binary vector whose element $[x_a^{(\omega)}]$ is equal to 1 iff actor a acts on the action area $S^{(\omega)}$ defined by event $\omega \in \Omega$;
- $\mathbf{V}^{(\omega)}$ is a vector whose element $[v_a^{(\omega)}]$ represents the velocity assigned to actor a;
- η_a^f is the *efficiency* of actor a acting on an event type $f \in \mathcal{F}^{(\omega)}$, i.e., the ratio between the effect produced by the action energy applied to the action area and the action energy itself;
- $E_a^{Av}[J]$ is the *available energy* of actor a evaluated at the instant when event ω occurs;
- $T^C[s]$ is the *coordination delay*, i.e., the time needed to process the event data, reconstruct the event itself, and select the team of actors by solving problem $\mathbf{P}_{\mathbf{All}}^{(\omega)}$; note that the coordination delay does not depend on the event;
- $\mathcal{S}_A^I \in \mathcal{S}_A$ is the subset of actors in *IDLE* state when event ω occurs, i.e., actors that have not been assigned to act on action areas associated with previously occurred events;
- N_S^a is the total number of sources sending packets to actor a, while $\Psi(N_S^a)[J]$ is a *penalty function* weighting the choice of actor a, which is receiving data from N_S^a sources, to be part of an acting team. The penalty function monotonically increases as N_S^a increases.

We now formulate the multiactor task allocation problem. $\mathbf{P}_{All}^{(\omega)}$: **Multiactor Task Allocation Problem**

$$Find: \mathbf{X}^{(\omega)} = \left[x_a^{(\omega)}\right],\ \mathbf{V}^{(\omega)} = \left[v_a^{(\omega)}\right]$$

$$Minimize: \sum_{a \in \mathcal{S}_a^I} x_a^{(\omega)} \cdot \left[E_a^{M,(\omega)} + E_a^{\Omega,(\omega)} + \Psi\left(N_S^a\right)\right]$$

$Subject\ to:$

$$E_a^{M,(\omega)} = \left[\beta v_a^{(\omega)\gamma} + P_{min}^M\right] \cdot T_a^{M,(\omega)},\ \forall a \in \mathcal{S}_a^I; \tag{14}$$

$$T_a^{M,(\omega)} = \frac{d_a^{(\omega)}}{v_a^{(\omega)}},\ \forall a \in \mathcal{S}_a^I; \tag{15}$$

$$v_a^{min} \le v_a^{(\omega)} \le v_a^{max},\ \forall a \in \mathcal{S}_a^I; \tag{16}$$

$$E_a^{\Omega,(\omega)} = l_a^f \cdot T_a^{\Omega,(\omega)} \ge 0,\ \forall a \in \mathcal{S}_a^I,\ f \in \mathcal{F}^{(\omega)}; \tag{17}$$

$$\sum_{a \in \mathcal{S}_a^I} x_a^{(\omega)} \cdot \eta_a^f \cdot E_a^{\Omega,(\omega)} \ge S^{(\omega)} \cdot I^{(\omega)},\ f \in \mathcal{F}^{(\omega)}; \tag{18}$$

$$T_a^{M,(\omega)} + T_a^{\Omega,(\omega)} \le D^{(\omega)} - T^C,\ \forall a \in \mathcal{S}_a^I; \tag{19}$$

$$E_a^{M,(\omega)} + E_a^{\Omega,(\omega)} \ge E_a^{Av},\ \forall a \in \mathcal{S}_a^I; \tag{20}$$

$$\sum_{a \in \mathcal{S}_a^{F,(\omega)}} x_a^{(\omega)} \ge 1. \tag{21}$$

Constraint (14) defines the energy required for actor a to move to the action area defined by the occurring event, which is the product of the power needed to move and the time needed to reach the action area at a given velocity; this time is expressed as the ratio between the distance of the actor from the action area and the selected velocity, as expressed in (15). Constraint (16) bounds the velocity range for each actor. Constraint (17) defines the energy required for actor a to complete the action when it is part of an acting team. Constraint (18) assures that the selected team be able to complete the assigned task, given the characteristics of the actor composing the team, and the scope and intensity of the event. Constraint (19) limits the sum of the action completion time and the time required to move the actor team to be smaller than the action completion bound, discounted by the coordination delay. Constraint (20) guarantees a nonnegative residual energy for each actor. Finally, constraint (21) ensures that at least one actor acts on the advertised action area.

Algorithm 4. Event preemption for multiactor task allocation

```
if ( P_All^(ω)/Ω_Active == FEASIBLE ) then

    SOLVE (P_All^(ω)/Ω_Active)
    Ω_Active ≡ Ω_Active ∪ ω
    UPDATE (S_A^I)
else
    σ_min = argmin_{σ∈Ω_Active} Pr^(σ)
    if ( Pr^(ω) > Pr^(σ_min) ) then
        Ω'_Active ≡ Ω_Active \ σ_min
        UPDATE (S_A^I)

        SOLVE (P_All^(ω)/Ω'_Active)
        Ω''_Active ≡ Ω'_Active ∪ ω

        UPDATE (S_A^I)
        if ( P_All^(σ_min)/Ω''_Active == FEASIBLE ) then
```

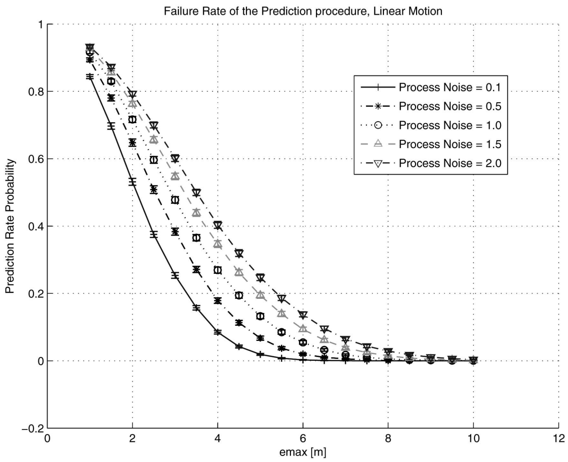

Fig. 3. Failure rate of the prediction procedure, with linear motion, for different levels of process noise.

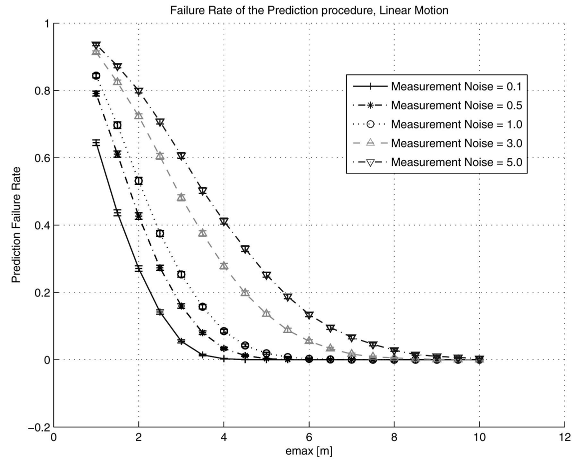

Fig. 4. Failure rate of the prediction procedure, with linear motion, for different levels of measurement noise.

$SOLVE\,(\mathbf{P}_{All}^{(\sigma_{min})}/\Omega''_{Active})$
$\Omega_{Active} \equiv \Omega''_{Active} \cup \sigma_{min}$
$UPDATE\,(\mathcal{S}_A^I)$
end if
end if
end if

Algorithm 4 defines the event-preemption policy for multiactor task allocation in the case where resources are insufficient to accomplish a high priority task. For the sake of simplicity, task reallocation is performed only if actions associated with higher priority events cannot be accomplished because of lack of resources, as it stated in the assumptions reported in this section. More specifically, if the task associated with event ω cannot be accomplished, given the resource already allocated to all active events (Ω_{Active}), i.e., if $\mathbf{P}_{All}^{(\omega)}/\Omega_{Active}$ is unfeasible, then Algorithm 4 proceeds with the preemption of all those ongoing tasks characterized by lower priorities, if any. The objective of this preemptive scheme is to reallocate useful resource to higher priority events that could not be successfully completed otherwise, while minimizing the number of costly task reallocations.

6 Performance Results

Section 6.1 discusses our proposed algorithms for sensor-actor communication, while Section 6.2 evaluates our actor-actor coordination scheme.

6.1 Sensor-Actor Communication

Performance results shown in this section are obtained with the sensor-actor simulator that we developed within the J-SIM framework [43]. First, we discuss results relevant to the prediction procedure described in Section 3. Actors move according to the model described in Section 3.2. In the first set of simulations, each actor selects a target destination and moves at constant speed to reach it. The actor implements a proportional controller that generates input commands to compensate for the process noise (random acceleration) by reestablishing the correct direction and speed. At each step, the actor measures its position (which is affected by measurement noise), filters the data, and decides whether an update needs to be sent.

In Figs. 3 and 4, we report the *failure rate* of the prediction procedure, with varying values for e_{max}, and for different values of noise. The failure rate is defined as the number of location updates sent over all measurements taken at the actor. Each figure reports results averaged over different simulation scenarios, with 95 percent confidence intervals. In Fig. 3, we report the failure rate with varying process noise, while in Fig. 4, we show the failure rate with varying measurement noise. In the range of values analyzed, which corresponds to realistic motion scenarios, it is shown that if it is possible to accept a localization error of $5\ \mathrm{m}$ for the actors, which is reasonable being around 10 percent of the transmission range, the prediction at the sensors allows the actor to avoid 75 percent and more location updates, with proportional energy savings at the sensors. In the second set of simulations, reported in Fig. 5, actors select several different destinations during each simulation, similarly to a (perturbed) Random Waypoint model. The failure rate is only slightly higher, which shows that the prediction procedure proposed is effective even

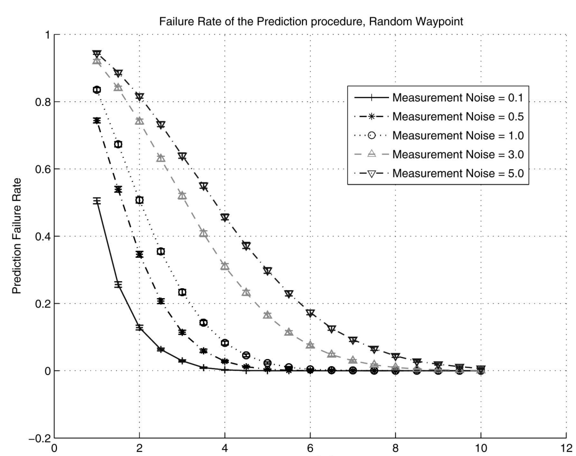

Fig. 5. Failure rate of the prediction procedure, with random waypoint motion, for different levels of measurement noise.

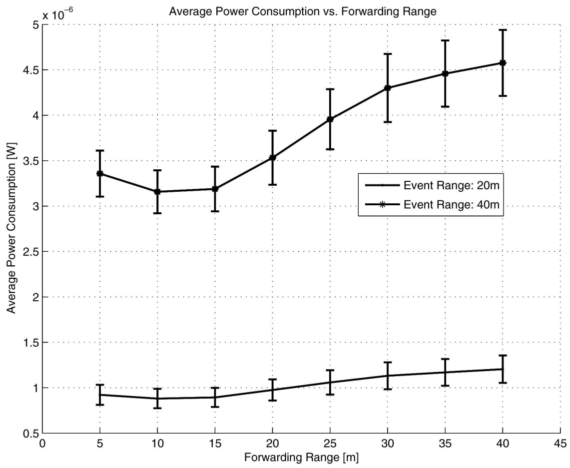

Fig. 6. Average power consumption versus forwarding range, low and moderate traffic.

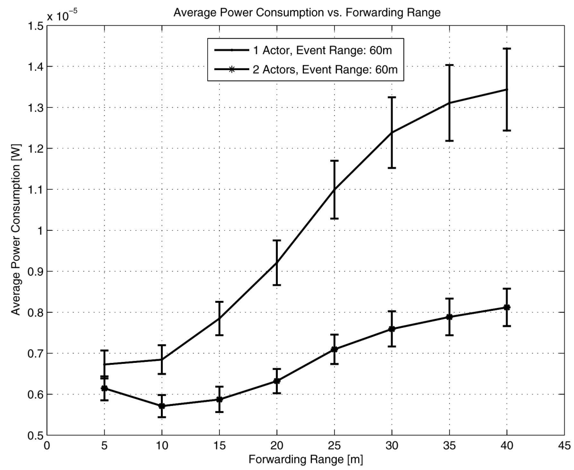

Fig. 8. Average power consumption versus forwarding range, high traffic.

when complex movement patterns are in place, and shows good robustness against noise.

As far as sensor-actor communication is concerned, sensors implement the geographical forwarding algorithm described in Section 4. The MAC layer is based on CSMA/CA. At the physical layer, we implemented our power control procedure and set bandwidth and power consumption parameters similar to IEEE 802.15.4 compliant radios according to the Texas Instruments/Chipcon CC2420 data sheet. The monitored area is a $200 \text{ m} \times 200 \text{ m}$ square, with 200 randomly deployed sensors. The maximum transmission range of sensors is set to 40 m and the bandwidth to 250 Kbit/s. Sensors send 56 byte long packets with a reporting rate of 1 packet/s, and the size of the queues is set to 20 packets. We perform terminating simulations that last 400 s, average over different random topologies, and show 95 percent confidence intervals.

In Figs. 6 and 7, we show a comparison of the average power consumption and delay, respectively, with increasing forwarding range. Sensors inside the event area report measurements to the actor. The *event area* is circular and centered at (100, 100) m. The figures report simulation runs for the cases of low and moderate traffic, i.e., the *event range* is equal to 20 and 40 m around the center, respectively. In the first case, on average, 7 sensors reside in the event area, while in the second case, there are around 25 sources. In Figs. 6 and 7, we show that in situations of low and moderate traffic, which are common in sensor networks, the end-to-end delay can be consistently decreased by increasing the forwarding range. This is an important trade-off that has not been thoroughly explored so far. Clearly, this is paid with increased power consumption with respect to the optimal values.

Fig. 8 refers to a high traffic scenario. The event range is set to 60 m, which corresponds to 57 sources, on average. The event area lies completely in the Voronoi cell of a single actor. We compare energy consumption, delay, and packet drops when one or two actors receive the traffic generated in the event area, i.e., with or without the congestion control procedure devised in Section 4.2. We observe the following behavior. In the first case (no congestion control), the event area itself is congested and a high percentage of packets are dropped (between 15 and 40 percent) (Fig. 10), while the end-to-end delays increase to about 1 s and are not easily controlled by changing the forwarding range (Fig. 9). Note that packets are dropped mostly in the event area due to the

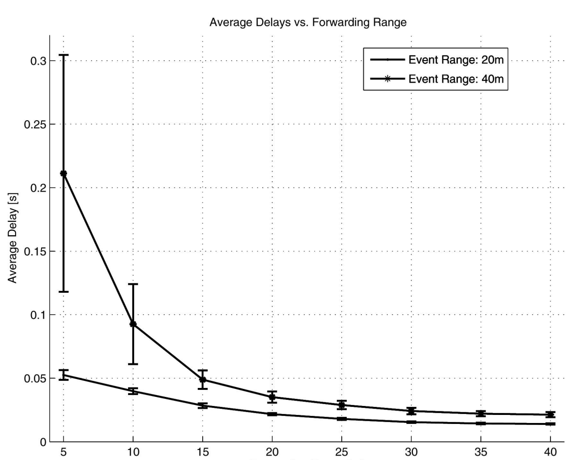

Fig. 7. Delay versus forwarding range, low and moderate traffic.

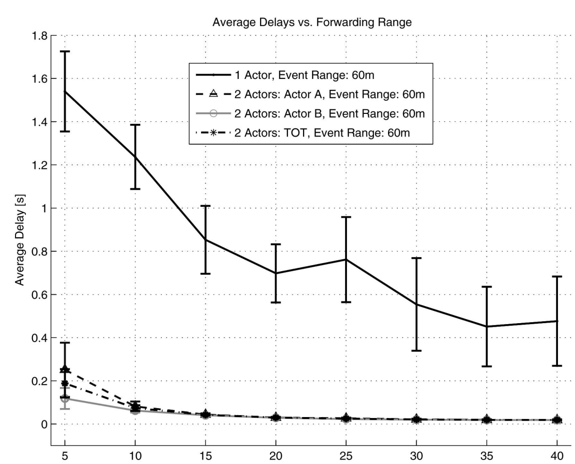

Fig. 9. Delay versus forwarding range, high traffic.

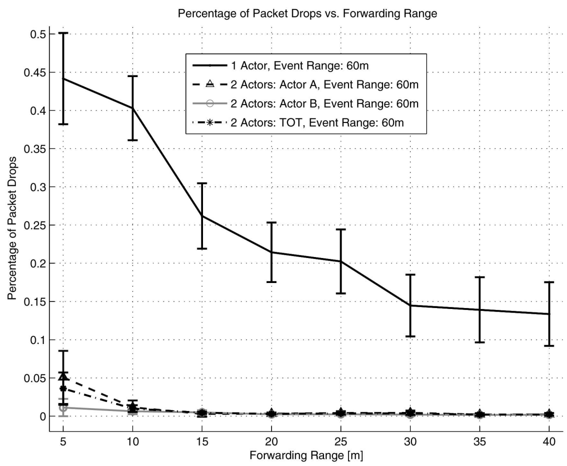

Fig. 10. Packet drops versus forwarding range, high traffic.

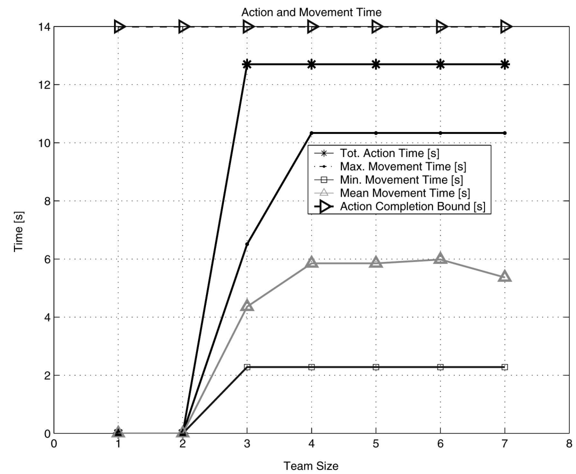

Fig. 12. Delay versus maximum team size.

multiple collisions at the MAC layer. Closer to the actor, the traffic is decreased due to earlier drops, and fewer nodes try to transmit simultaneously. Conversely, congestion can dramatically be decreased when the proposed congestion control procedure divides the event data between two actors. This is due to the fact that most of the congestion and packet drops occur in the event area, where many nodes try to transmit simultaneously, with the consequent drops due to simultaneous transmissions. This is dramatically improved when a second actor on the opposite side of the event area receives data, since traffic is diverted from the event area. The percentage of packets dropped is close to nil (see Fig. 10), delays are two orders of magnitude lower and can be regulated with power control (Fig. 9). Importantly, even though the second actor is farther (thus, in theory, suboptimal) from the event area, and although without congestion control, packets are dropped early on their source-actor path, the power consumption is also decreased by the congestion control procedure, mostly due to the reduced packet retransmissions at the MAC layer (Fig. 8).

6.2 Actor-Actor Coordination

In this section, we discuss performance results for the multiactor task allocation problem presented in Section 5. In the simulations performed, actors are assumed to be randomly deployed in a $200\ \mathrm{m} \times 200\ \mathrm{m}$ area, where events with intensity $I = 0.5\ \mathrm{J/m^2}$ and scope $S = \pi \cdot 4^2\ \mathrm{m^2}$ occur randomly in the entire area. Actors are assumed to be randomly deployed in a $200\ \mathrm{m} \times 200\ \mathrm{m}$ area, where events with intensity $I = 0.5\ \mathrm{J/m^2}$ and scope $S = \pi \cdot 4^2\ \mathrm{m^2}$ occur randomly in the entire area. We set the action completion bound D and the coordination delay T^C to 15 and 1 s, respectively. We consider a scenario with homogeneous actors, with $\beta = 0.05\ \mathrm{W/(m/s)}^\gamma, \gamma = 1.5, P^M_{min} = 1\ \mathrm{W}$, efficiency $\eta = 1$, action power $l = 1\ \mathrm{W/m^2}$, and initial energy $E_0 = 1{,}000$ J; moreover, the velocities range in the interval $[3, 12]\ \mathrm{m/s}$.

Figs. 11 and 12 report results from a set of simulations, where we impose a limit on the maximum team size, i.e., the maximum number of actors taking part in an acting team, reported on the x-axis; while in Fig. 13, the number of actors composing a team is forced to be fixed and equal to the team size, which is reported on the x-axis. Interestingly, when the number of actors taking part in an acting team is optimized to minimize the overall energy expenditure, i.e., the sum of the movement energy E^M and the action energy E^Ω, at least three actors are needed to complete the action

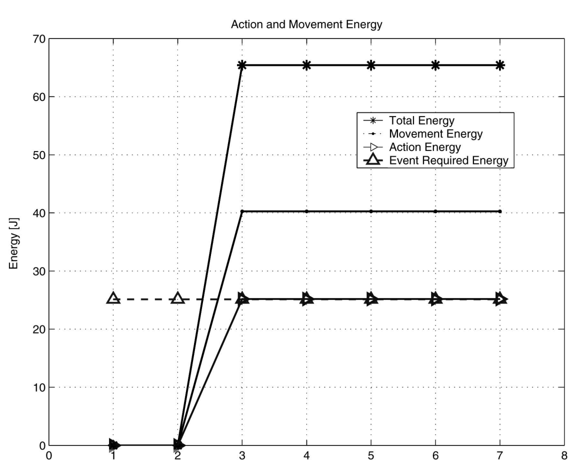

Fig. 11. Energy consumption versus maximum team size.

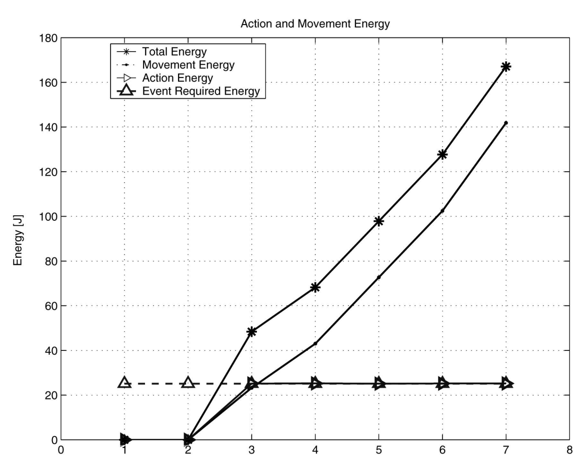

Fig. 13. Energy consumption versus team size.

(see Fig. 11) and the total action time tends to be exactly the maximum allowed completion bound D, discounted by the coordination delay T^C (see Fig. 12). Problem $\mathbf{P}_{All}^{(\omega)}$ tends to minimize the number of involved actors and assign higher speed to those actors that are closer to the action area. This can be explained by considering that a fixed amount of power (P_{min}^{M}) is dissipated every time an actor needs to move, irrespective of its velocity. Conversely, when all the available actors are forced to be part of a team, the action time can be reduced at the expense of energy consumption, as reported in Fig. 13.

7 Conclusions

We discussed challenges for coordination and communication in WSANs with mobile actors, and presented effective solutions for the sensor-actor and actor-actor coordination problems. First, we proposed a proactive location management scheme to handle the mobility of actors with minimal energy expenditure for sensors. The scheme enables geographical routing, based on which an energy-efficient communication solution was derived for sensor-actor communication. We showed how to control the delay of the data-delivery process based on power control, and how to deal with network congestion by forcing multiple actors to share the traffic generated in the event area. Finally, a model for actor-actor coordination was introduced that coordinates motion based on the characteristics of the event.

Acknowledgments

A preliminary shorter version of this paper [1] was presented at the IEEE SECON 2006, Reston, Virginia. This material is based upon work supported by the US National Science Foundation under Grant ECCS-0701559.

References

[1] T. Melodia, D. Pompili, and I.F. Akyildiz, "A Communication Architecture for Mobile Wireless Sensor and Actor Networks," *Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* Sept. 2006.

[2] I.F. Akyildiz and I.H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks,* vol. 2, no. 4, pp. 351-367, Oct. 2004.

[3] G. Wang, G. Cao, T.L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," *Proc. IEEE INFOCOM,* Mar. 2005.

[4] P. Ogren, E. Fiorelli, and N.E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Trans. Automatic Control,* vol. 49, no. 8, pp. 1292-1302, Aug. 2004.

[5] R. Shah, S. Roy, S. Jain, W. Brunette, and G. Borriello, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Ad Hoc Networks,* vol. 1, no. 3, pp. 215-233, Sept. 2003.

[6] M. Rahimi, H. Shah, G. Sukhatme, J. Heidemann, and D. Estrin, "Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network," *Proc. IEEE Int'l. Conf. Robotics and Automation,* pp. 19-24, May 2003.

[7] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. ACM MobiCom,* Sept. 2004.

[8] T. Melodia, D. Pompili, V.C. Gungor, and I.F. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks," *IEEE Trans. Mobile Computing,* vol. 6, no. 10, pp. 1116-1129, Oct. 2007.

[9] J. Li, J. Jannotti, D.D. Couto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom,* pp. 120-30, 2000.

[10] S.M. Das, H. Pucha, and Y.C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," *Proc. IEEE INFOCOM,* Mar. 2005.

[11] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks," *Proc. ACM Conf. Embedded Networked Sensor Systems (SENSYS),* Nov. 2004.

[12] M. Rossi and M. Zorzi, "Cost Efficient Localized Geographical Forwarding Strategies for Wireless Sensor Networks," *Proc. Tyrrhenian Int'l Workshop Digital Comm. (TIWDC '05),* July 2005.

[13] J.A. Stankovic, T.F. Abdelzaher, C. Lu, L. Sha, and J. Hou, "Real-Time Communication and Coordination in Embedded Sensor Networks," *Proc. IEEE,* vol. 91, no. 7, pp. 1002-1022, July 2003.

[14] R. Vedantham, Z. Zhuang, and R. Sivakumar, "Hazard Avoidance in Wireless Sensor and Actor Networks," *Computer Comm.,* vol. 29, nos. 13/14, pp. 2578-2598, Aug. 2006.

[15] R. Vedantham, Z. Zhuang, and R. Sivakumar, "Mutual Exclusion in Wireless Sensor and Actor Networks," *Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* Sept. 2006.

[16] A. Durresi, V. Paruchuri, and L. Barolli, "Delay-Energy Aware Routing Protocol for Sensor and Actor Networks," *Proc. Int'l. Conf. Parallel and Distributed Systems,* vol. 1, pp. 292-298, July 2005.

[17] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Real-Time Routing Protocol for Sensor Networks," *Proc. IEEE Int'l. Conf. Distributed Computing Systems (ICDCS),* pp. 46-55, May 2003.

[18] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* Mar. 2005.

[19] T. Melodia, D. Pompili, and I.F. Akyildiz, "On the Interdependence of Distributed Topology Control and Geographical Routing in Ad Hoc and Sensor Networks," *J. Selected Areas Comm.,* vol. 23, no. 3, pp. 520-532, Mar. 2005.

[20] F. Aurenhammer, "Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys,* vol. 23, pp. 345-405, 1991.

[21] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Networks," *Proc. IEEE INFOCOM,* Apr. 2001.

[22] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks," *Proc. ACM MobiCom,* pp. 139-150, 2001.

[23] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Worst and Best-Case Coverage in Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 4, no. 1, pp. 84-92, Jan./Feb. 2005.

[24] I. Stojmenovic, A.P. Ruhil, and D.K. Lobiyal, "Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks," *Wiley Wireless Comm. and Mobile Computing,* vol. 6, pp. 247-258, 2006.

[25] S. Tilak, V. Kolar, N.B. Abu-Ghazaleh, and K.-D. Kang, "Dynamic Localization Control for Mobile Sensor Networks," *Proc. IEEE Int'l Performance Computing and Comm. Conf. (IPCCC),* Apr. 2005.

[26] A. Deshpande and S. Madden, "Mauvedb: Supporting Model-Based User Views in Database Systems," *Proc. ACM SIGMOD,* pp. 73-84, 2006.

[27] R. Olfati-Saber, "Distributed Tracking for Mobile Sensor Networks with Information-Driven Mobility," *Proc. Am. Control Conf.,* July 2007.

[28] A. Ahmad, M. Gani, and F. Yang, "Decentralized Robust Kalman Filtering for Uncertain Stochastic Systems over Heterogeneous Sensor Networks," *Signal Processing,* vol. 88, no. 8, pp. 1919-1928, 2008.

[29] H. Medeiros, J. Park, and A. Kak, "Distributed Object Tracking Using a Cluster-Based Kalman Filter in Wireless Camera Networks," *IEEE J. Selected Topics in Signal Processing,* vol. 2, no. 4, pp. 448-463, Aug. 2008.

[30] I.N. Psaromiligkos and S.N. Batalama, "Recursive Short-Data-Record Estimation of AV and MMSE/MVDR Linear Filters for DS-CDMA Antenna Array Systems," *IEEE Trans. Comm.,* vol. 52, no. 1, pp. 136-148, Aug. 2004.

[31] P.S. Maybeck, *Stochastic Models, Estimation, and Control,* vol. 1. Academic Press, 1979.

[32] I. Rekleitis, "A Particle Filter Tutorial for Mobile Robot Localization," Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill Univ., Montreal, 2004.

[33] R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering,* third ed. John Wiley & Sons, Inc., 1996.

[34] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, pp. 51-58, May 2000.
[35] *Crossbow MICAz Mote Specifications,* http://www.xbow.com, 2009.
[36] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *ACM Wireless Networks,* vol. 7, no. 6, pp. 609-616, Nov. 2001.
[37] M. Zuniga and B. Krishnamachari, "Analyzing the Transitional Region in Low Power Wireless Links," *Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* Sept. 2004.
[38] C. Li, W. Hsu, B. Krishnamachari, and A. Helmy, "A Local Metric for Geographic Routing with Power Control in Wireless Networks," *Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* Sept. 2005.
[39] Y.T. Hou, Y. Shi, and H.D. Sherali, "Optimal Base Station Selection for Anycast Routing in Wireless Sensor Networks," *IEEE Trans. Vehicular Technology,* vol. 55, no. 3, pp. 813-821, May 2006.
[40] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Comm.,* vol. 1, no. 4, pp. 660-670, Oct. 2002.
[41] O. Akan and I.F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 13, no. 5, pp. 1003-1017, Oct. 2005.
[42] B.P. Gerkey and M.J. Mataric, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The Int'l J. Robotics Research,* vol. 23, no. 9, pp. 939-954, Sept. 2004.
[43] *The J-Sim Simulator,* http://www.j-sim.org/, 2009.

Tommaso Melodia received the laurea (integrated BS and MS) and doctorate degrees in telecommunications engineering from the University of Rome La Sapienza, Italy, in 2001 and 2005, respectively, and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in June 2007. He is an assistant professor in the Department of Electrical Engineering at the University at Buffalo, the State University of New York (SUNY), where he directs the Wireless Networks and Embedded Systems Laboratory. He is the recipient of the BWN-Lab Researcher of the Year Award for 2004. He coauthored a paper that was recognized as the *Fast Breaking Paper in the field of Computer Science* for February 2009 by Thomson ISI Essential Science Indicators. He is an associate editor of the *Computer Networks* (Elsevier) journal, *Transactions on Mobile Communications and Applications* (ICST), and the *Journal of Sensors* (Hindawi). He serves on the technical program committees of several leading conferences in wireless communications and networking, including IEEE Infocom, ACM Mobicom, and ACM Mobihoc. He is the technical cochair of the Ad Hoc and Sensor Networks Symposium for IEEE ICC 2009. His current research interests include modeling and optimization of multihop wireless networks, cross-layer design and optimization, wireless multimedia sensor and actor networks, underwater acoustic networks, and cognitive radio networks. He is a member of the IEEE.

Dario Pompili received the laurea (integrated BS and MS) and doctorate degrees in telecommunications engineering and system engineering from the University of Rome La Sapienza, Italy, in 2001 and 2004, respectively, and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech) in June 2007, after working at the Broadband Wireless Networking Laboratory (BWN-Lab) under the direction of Professor Ian F. Akyildiz. He joined the Faculty of the Department of Electrical and Computer Engineering at Rutgers University as an assistant professor in Fall 2007, where he is now the director of the Cyber-Physical Systems Laboratory (CPS-Lab). In 2005, he was awarded the Georgia Tech BWN-Lab Researcher of the Year for "outstanding contributions and professional achievements." His research interests include ad hoc and sensor networks, underwater acoustic communications, wireless sensor and actor networks, and network optimization and control. He is the author and coauthor of several influential research articles on these topics. He serves on the editorial board of the *Ad Hoc Networks* (Elsevier) journal and the technical program committee of several leading conferences on networking. He is a member of the IEEE, the IEEE Communications Society, and the ACM.

Ian F. Akyildiz is the Ken Byers distinguished chair professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech). Since June 2008, he has been an honorary professor in the School of Electrical Engineering at the Universitat Politecnico de Catalunya, Barcelona, Spain. He is the editor-in-chief of the *Computer Networks* (COMNET) journal as well as the founding editor-in-chief of the *Ad Hoc Networks* journal and *Physical Communication* journal, all with Elsevier. His current research interests are in cognitive radio networks, wireless sensor networks, and nanocommunication networks. He has received numerous awards, including the 1997 IEEE Leonard G. Abraham Prize Award (the IEEE Communications Society) for his paper entitled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the *IEEE JSAC* in January 1996; the 2002 IEEE Harry M. Goode Memorial Award (the IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking"; the 2003 IEEE Best Tutorial Award (the IEEE Communication Society) for his paper entitled "A Survey on Sensor Networks" published in the *IEEE Communications Magazine* in August 2002; the 2003 ACM Sigmobile Outstanding Contribution Award with the citation "for pioneering contributions in the area of mobility and resource management for wireless communication networks"; the 2004 Georgia Tech Faculty Research Author Award for his "outstanding record of publications of papers between 1999 and 2003"; the 2005 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech; the 2009 Georgia Tech Outstanding Doctoral Thesis Advisor Award for his 20+ years of service and dedication to Georgia Tech and producing outstanding PhD students; and the 2009 Electrical and Computer Engineering Distinguished Mentor Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been a fellow of the ACM since 1996. He is a fellow of the IEEE.