This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE SECON 2006 proceedings.

A Communication Architecture for Mobile Wireless Sensor and Actor Networks

Tommaso Melodia, Dario Pompili, Ian F. Akyildiz

Broadband and Wireless Networking Laboratory
Georgia Institute of Technology, Atlanta, GA 30332
e-mail: {tommaso, dario, ian}@ece.gatech.edu

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.**

In this paper, the coordination and communication problems in WSANs with mobile actors are studied. A hybrid location management scheme is introduced to handle the mobility of actors with minimal energy expenditure. Actors broadcast location updates limiting their scope based on Voronoi diagrams, whereas sensors predict the movement of actors based on Kalman filtering of previously received updates. An optimal energy-aware forwarding rule is then derived for sensor-actor communication, based on geographical routing. The proposed scheme allows controlling the delay of the data-delivery process based on power control, and deals with network congestion by forcing multiple actors to be recipients for traffic generated in the event area. The motion of actors is coordinated to optimally accomplish the tasks based on the characteristics of the events.

I. Introduction

THE convergence of communication and computation with signal processing and several branches of control theory such as robotics and artificial intelligence is enabling distributed systems of embedded devices that sense, interact, and control the physical environment. Wireless Sensor and Actor[1] Networks (WSANs) [1] are distributed wireless systems of heterogeneous devices referred to as *sensors* and *actors*. 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.

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 [2], enable adaptive sampling of the environment [3], pick up data from the sensors when in close range, buffer it, and drop off the data to wired access points [4], or perform energy harvesting [5].

[1]It may be worth specifying how the term *actor* differs from the more conventional notion of *actuator*. From our perspective an actor, besides being able to act on the environment by means of several actuators, is also a *single network entity* that performs networking-related functionalities, i.e., receive, transmit, and relay data. For example, the mobility of a robot may be enabled by several motors and servo-mechanisms (actuators). However, from a networking perspective, the robot constitutes a single entity, which we refer to as actor.

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 [1]. 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 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 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.

In our prior work on WSANs [6], we proposed 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 [6]. 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 expenditure for the sensors. The proposed solution is tailored for WSAN applications and overcomes the drawbacks of previously proposed localization services [7][8]. 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 is shown to consistently reduce the energy consumption on sensors by avoiding over 75% of location updates with respect to existing location update algorithms.

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. We derive a simple yet optimal forwarding rule based on geographic position in presence of Rayleigh fading channels. With respect to previously proposed geographic forwarding rules [9][10], 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, it selects the best actor(s) to form the actor team to perform the required actions, based on the characteristics of the event, and drives the motion of the team towards the relevant area.

The paper is organized as follows. In Section II, we describe the proposed location management scheme, while in Section III, we describe the sensor-actor communication solution. In Section IV, we introduce the actor-actor coordination model. In Section V, we present performance evaluation results, while in Section VI we conclude the paper.

II. 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 rate wireless interface for actor-actor communication. From the perspective of sensors, actors are *equivalent recipients of information*. Hence, each sensor will try to 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 [6][11][10][9], we study the *sensor-actor coordination* based on a geographical routing paradigm. Geographical routing algorithms are attractive especially for their *scalability*, as it is possible to scale the network size without increasing the signaling overhead, because routing decisions are inherently *localized* [11]. 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 availability of efficient location management schemes that are able to 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 [7], 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. The location servers for each node are determined based on a hashing function in the node identifier space. In [8], the performance of GLS is compared to two other location services based on similar premises. In general, the objective of these mechanisms, which can be classified as rendezvous-based protocols [8], is to potentially allow each single device in the network to retrieve the location of any other node, based on queries and replies. Clearly, query-based mechanisms can introduce delays that may not be acceptable in delay-critical systems such as WSANs. Moreover, the extensive message exchange and complex server structures, often hierarchical, associated with these protocols, can be avoided given the characteristics of WSANs.

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 [12]. 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, will need to update their position much less frequently than actors that follow temporally uncorrelated trajectories.

A. Limiting Broadcasts in Space

As discussed, 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 [13], they are used along with Delaunay triangulation to study sensor network coverage. Instead, we leverage Voronoi diagrams to limit the spatial extension of actor broadcasts.

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 from. With respect to flooding, the energy consumption for location updates is drastically reduced. It can be shown that 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 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 actors.

B. Limiting Broadcasts in Time

The dynamic movement model for the i^{th} 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 [14] 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 non-idealities 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 $\mathbf{H} = \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix}$, $\mathbf{C} = \mathbf{B}$.

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 [15], etc.

The Kalman filter [16] provides a computationally efficient set of recursive equations to estimate the state of such process, and can be proven to be the optimal filter in the minimum square sense. 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

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

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.,

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

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

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

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 size of the state space is only 4. Moreover, the processing cost for sensors is much lower than the communication cost. This is justified by [17], where the energy necessary to transmit 1 kbit is shown to be equivalent to the energy necessary to execute 300,000 processor instructions.

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 $(\mathbf{z_i^k} - \mathbf{H}\mathbf{\hat{x}_{i,s}^{k-}}) > e_{max}$. If sensor s does not receive a location update at step k, it assumes $\mathbf{z_i^k} = \mathbf{H}\mathbf{\hat{x}_{(i,s)}^{k-}}$, 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).

III. Sensor-Actor Communication

In [6], 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 pre-defined 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 [2], 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

[2]Whenever a packet is dropped by an intermediate sensor, either because it violates the latency bound constraint or because of network or channel impairments, the actor is notified so that the lost packet can be taken into account in the computation of the reliability.

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 has to be delivered within a given time bound B with a given reliability r_{th}. The solution presented in [6], 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 III-A. In case of congestion at a recipient actor, we re-route part of the traffic to another, less congested, actor.

A. Power-controlled Energy-delay Adjustment

Previous work on geographical routing considered primarily greedy forwarding[3] 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 transmission range. When a realistic physical layer is considered, such links are likely to be unstable and prone to high packet error rates. Hence, [10][9] propose enhanced flavors of greedy forwarding that avoid using those links. However, the objective is still to maximize the advance towards 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 [10][9], we remove the *unit disk graph* assumption relied on by most routing research, and consider a more accurate connectivity model. Hence, we first derive the energy-efficient forwarding distance in the presence of a fast fading channel. Then, we propose a mechanism to decrease the end-to-end delay by increasing the transmit power.

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

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

where P_{ij}^t is the power transmitted at v_i, Γ is a technology-dependent parameter representing the receiver threshold, and f is a unit-mean Rayleigh distributed r.v. that models fast fading for a given packet. Hence, we can write

$$\mathcal{P}_{ij}^s = \Pr\left\{f \geq \frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}\right\} = \int_{\frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}}^{+\infty} p_f(f)df = e^{-\frac{\pi}{4}\left(\frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}\right)^2}. \tag{9}$$

[3]Greedy forwarding has been enhanced in [18] by introducing face/perimeter routing techniques to route packets around the void area to reach the destination. This techniques can be applied to the mechanism proposed in this paper in low-density or concave areas.

The transmit power P_{ij}^t is related to the distance-dependent energy consumption through the transmit rate b as $P_{ij}^t = E_{\mathrm{amp}} \cdot d_{ij}^\alpha \cdot b$. We can interpret $E_{\mathrm{amp}} \cdot d_{ij}^\alpha \cdot b$ as the power necessary to transmit a packet 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 packet error rate as follows

$$P_{ij}^t = (E_{\mathrm{marg}} + E_{\mathrm{amp}}) \cdot d_{ij}^\alpha \cdot b. \tag{10}$$

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. This is expressed by

$$\mathcal{N}_{ij}^R(d, E_{\mathrm{marg}}) = \frac{1}{\mathcal{P}_{ij}^s} = e^{-\frac{\pi}{4}\left[\frac{\Gamma}{(E_{\mathrm{marg}}+E_{\mathrm{amp}})\cdot b}\right]^2}. \tag{11}$$

Now, consider a node v_i forwarding a packet towards a destination actor a_k at distance D. 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 $[J/bits]$. The end-to-end energy consumption can then be expressed as

$$E_{\mathrm{e-e}} = \sum_{(i,j)\in\mathcal{P}(v_i,a_k)} \left(\frac{P_{ij}^t}{r} + 2E_{\mathrm{elec}}\right), \tag{12}$$

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 plugging (10) in (12), and by considering retransmissions, we obtain

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

where $\mathcal{N}_{ij}^R$ is given by (11). The values for (d, E_{marg}) that minimize the above expression can be found by solving the nonlinear system $\nabla E_{\mathrm{e-e}} = \mathbf{0}$, i.e., $[\frac{\partial E_{\mathrm{e-e}}}{\partial d}, \frac{\partial E_{\mathrm{e-e}}}{\partial E_{\mathrm{marg}}}] = [0, 0]$, to find the stationary points of the function. A sufficient condition for a stationary point to be a a minimum is that the Hessian $\nabla^2 E_{\mathrm{e-e}}$ 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 consumption, and lends itself well to the development of localized forwarding rules. In case of ideal channel, and with $E_{\mathrm{marg}} = 0$, (13) is minimized when $d^{\mathrm{opt}} = \sqrt[\alpha]{\frac{2 \cdot E_{\mathrm{elec}}}{E_{\mathrm{amp}}(\alpha-1)}}$. With the parameters given in [19], i.e., $E_{\mathrm{elec}} = 50nJ/bit$, $E_{\mathrm{amp}} = 100pJ/bit/m^\alpha$, $\alpha = 2.5$, the optimal forwarding distance for an ideal channel is $d^{\mathrm{opt}} = 13.47m$. Solving (13) yields $d^{\mathrm{opt}} = 8.00m$ and $E_{\mathrm{marg}}^{\mathrm{opt}} = 86pJ/bit/m^\alpha$, i.e., $E_{\mathrm{marg}}^{\mathrm{opt}} \approx E_{\mathrm{amp}}$. Hence, as expected the optimal forwarding distance on a Rayleigh fading channel is lower than with an ideal channel, and a higher transmitting 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, it can be demonstrated that for values of α (path loss exponent) higher than 3.5, the expected energy consumption increases excessively when the next hop is closer than the optimal forwarding point to the destination. Hence, in this case, the next hop is selected as the closest node to the optimal forwarding point, among those that are not closer to the destination than the optimal forwarding point.

The reliability can be controlled 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 until the optimal forwarding range is reached. Transmitting closer than the optimal forwarding range, as will be shown in Section V-A, 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 III-B.

B. Network Layer Actor-driven Congestion Control

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 [6], 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 re-route 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 which 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, c_Δ. As a design choice, we set $c_\eta \geq c_\delta \geq c_\Delta$.

1) *Congestion factor* η_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 margin on the reliability to avoid instability.

2) *Directivity factor* δ_k, that 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. 1, 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 the example given in Fig. 1, 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. 1). 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

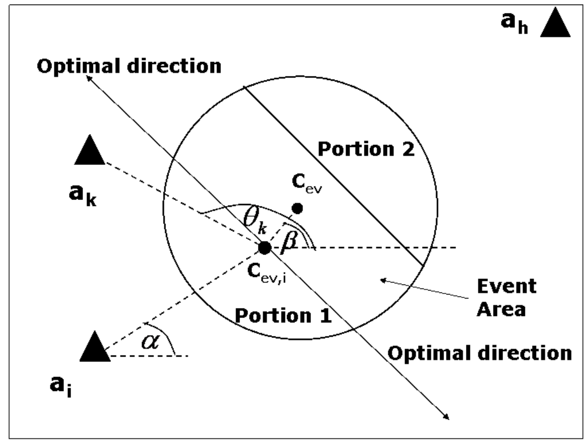

Fig. 1. Calculation of the directivity factor δ_i.

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

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., $\Delta_k = 1$ when the distance is maximal.

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* $\mathbf{x}_{\mathbf{k}*}^{\mathtt{virt}}$ 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 $\mathbf{x}_{\mathbf{k}*}$ 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 $\mathbf{x}_{\mathbf{k}*}^{\mathtt{virt}}$, while the real position $\mathbf{x}_{\mathbf{k}*}$ 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 in two.

IV. Actor-Actor Coordination

In this section, we formulate the multi-actor task allocation problem, whose objective is to coordinate the mobility. In particular, it selects the best actor(s) to form the *actor team*, and to control their motion toward the action area. Our previous work [6] 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 [6] 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.

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 multi-actor task allocation problem. In particular, the multi-actor 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 for 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 multi-actor allocation problem as a *Mixed Integer Non-Linear Program* (MINLP).

We define the problem according to the following assumptions: i) the energy to perform the action (action and movement energy) is orders of magnitude higher than the energy required for communication; ii) task reallocation is performed only if actions associated with higher priority events cannot be accomplished due to lack of resources.

We introduce the following notation:

- $l_a^f\,[W]$ 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 $\mathbf{P}_{All}^{(\omega)}$; 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 multi-actor task allocation problem.

$\mathbf{P}_{All}^{(\omega)}$: **Multi-actor Task Allocation Problem**

$$Find: \quad \mathbf{X}^{(\omega)} = [x_a^{(\omega)}],\ \mathbf{V}^{(\omega)} = [v_a^{(\omega)}]$$
$$Minimize: \quad \textstyle\sum_{a \in \mathcal{S}_a^I} x_a^{(\omega)} \cdot [E_a^{M,(\omega)} + E_a^{\Omega,(\omega)} + \Psi(N_S^a)]$$

$$Subject\ to:$$

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

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

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

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

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

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

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

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

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 non-negative residual energy for each actor. Finally, constraint (21) ensures that at least one actor act on the advertised action area.

Our actor-actor coordination mechanism includes an event-preemption policy for multi-actor task allocation for cases where resources are insufficient to accomplish a high priority task, which is omitted for lack of space.

V. Performance Results

Section V-A discusses our proposed algorithms for sensor-actor communication, while Section V-B evaluates our actor-actor coordination scheme.

A. *Sensor-actor Communication*

Performance results shown in this section are obtained with the sensor-actor simulator that we developed within the J-SIM framework [20]. First, we discuss results relevant to the prediction procedure described in Section II. Actors move according to the model described in Section II-B. 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. 2 and 3 we report the *failure rate* of the prediction procedure, with varying values for e_{max}, and for different values of the process noise (random acceleration). 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% confidence intervals. In Fig. 2 we report the failure rate with varying process noise, while in Fig. 3 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% of the transmission range, the prediction at the sensors allows the actor to avoid 75% and more location updates, with proportional energy savings at the sensors. In the second set of simulations, reported in Fig. 4, 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 when complicated movement patterns are in place, and shows good robustness properties against noise.

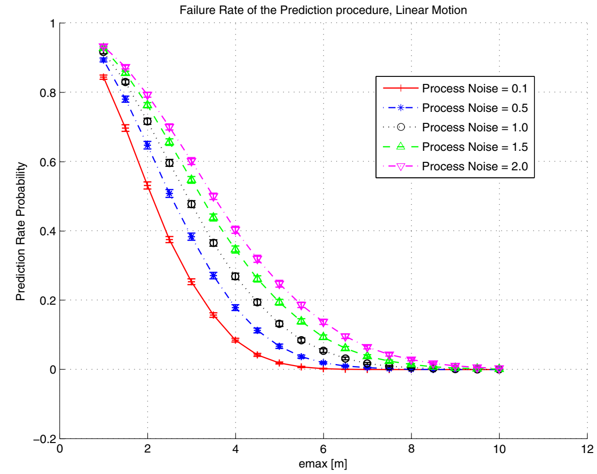

Fig. 2. Failure rate of the prediction procedure, with linear motion, for different levels of process noise.

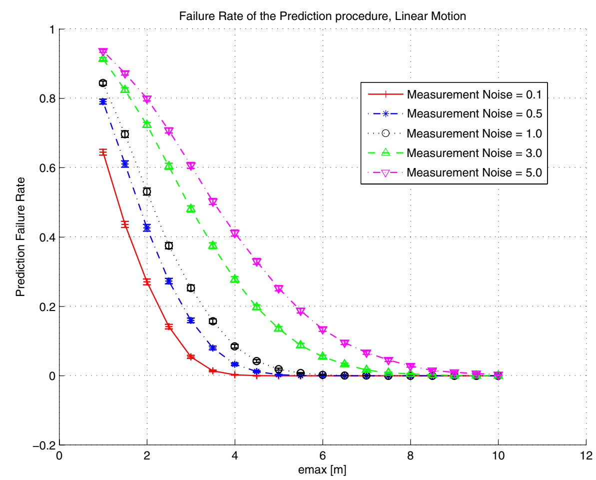

Fig. 3. Failure rate of the prediction procedure, with linear motion, for different levels of measurement noise.

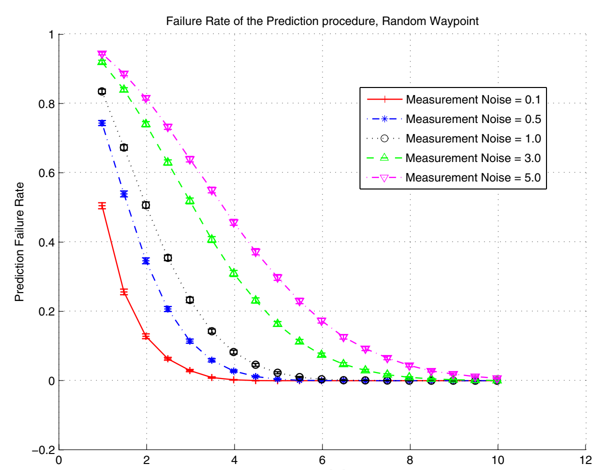

Fig. 4. Failure rate of the prediction procedure, with random waypoint motion, for different levels of measurement noise.

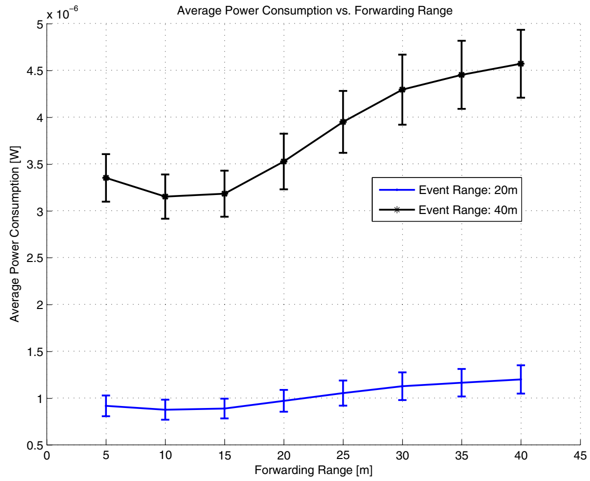

Fig. 5. Average power consumption vs. forwarding range, low and moderate traffic.

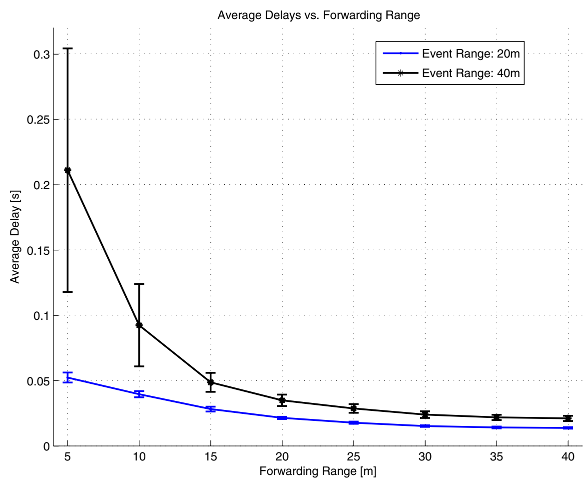

Fig. 6. Delay vs. forwarding range, low and moderate traffic.

As far as sensor-actor communication is concerned, sensors implement the geographical forwarding algorithm described in Section III. 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 Chipcon CC2420 datasheet. The monitored area is a $200\,\mathrm{m}$x$200\,\mathrm{m}$ square, with 200 randomly deployed sensors. The maximum transmission range of sensors is set to $40\,\mathrm{m}$, and the bandwidth to $250\,\mathrm{kbit/s}$. Sensors send $56\,\mathrm{byte}$ long packets with a reporting rate of $1\,\mathrm{packet/s}$, and the size of the queues is set to 20 packets. We perform terminating simulations that last $400\,\mathrm{s}$, average over different random topologies, and show 95% confidence intervals. The simulation parameters are reported in Table I.

In Figs. 5 and 6, 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 $(100, 100)\,\mathrm{m}$. The figures report simulation runs for the cases of low and moderate traffic, i.e, the *event range* is equal to $20\,\mathrm{m}$ and $40\,\mathrm{m}$ 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. 5 and 6 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.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
area	200x$200\,\mathrm{m}$
sensors	200
max tx range	$40\,\mathrm{m}$
bandwidth	$250\,\mathrm{kbit/s}$
packet size	$56\,\mathrm{bytes}$
queue	$20\,\mathrm{packets}$
reporting rate	$1\,\mathrm{packet/s}$
simulation time	$400\,\mathrm{s}$

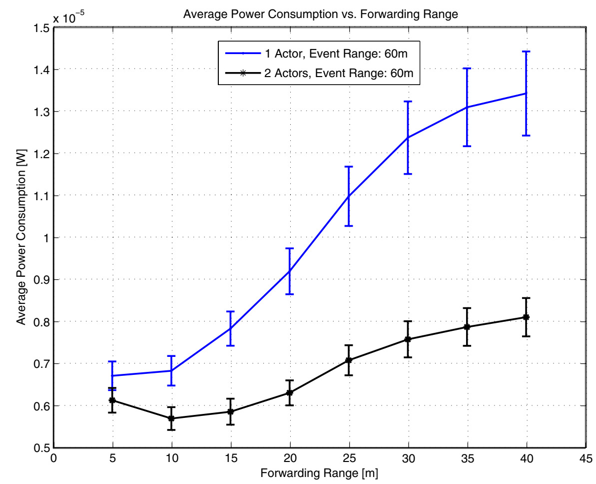

Fig. 7. Average power consumption vs. forwarding range, high traffic.

Figure 7 refers to a high traffic scenario. The event range is set to $60\,\mathrm{m}$, 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 1 or 2 actors receive the traffic generated in the event area, i.e., with or without the congestion control procedure devised in Section III-B. 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%) (Fig. 9), while the end-to-end delays increase to about $1\,\mathrm{s}$ and are not easily controlled by changing the forwarding range (Fig. 8). Note that packets are dropped mostly in the event area due to multiple collisions at the MAC layer. Closer to the actor, the traffic is decreased due to

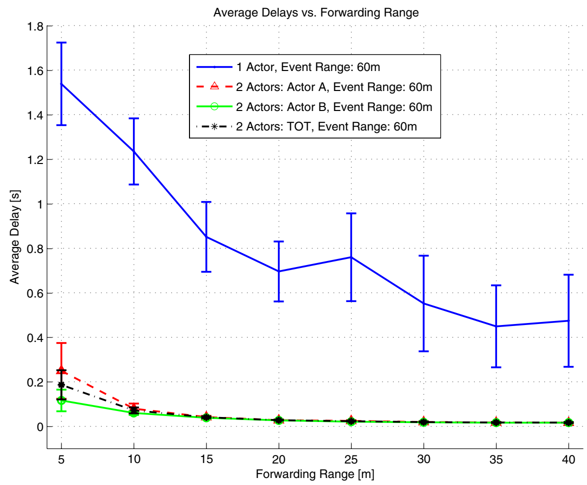

Fig. 8. Delay vs. forwarding range, high traffic.

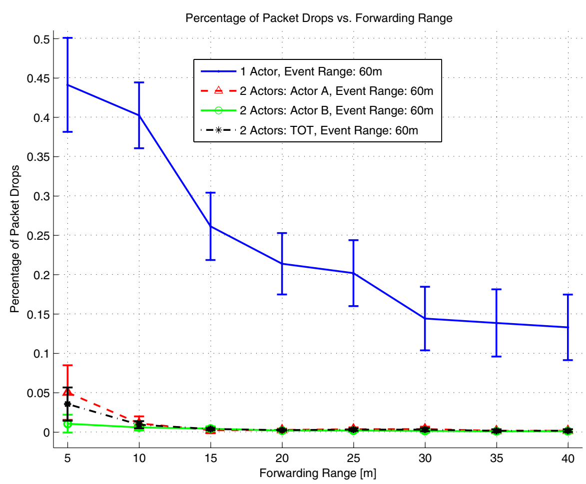

Fig. 9. Packet drops vs. forwarding range, high traffic.

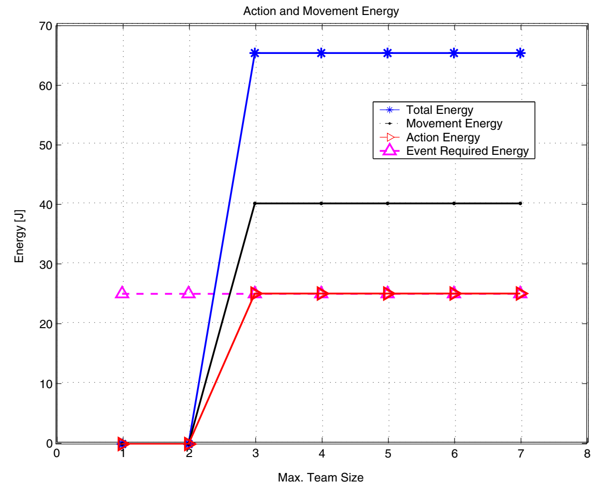

Fig. 10. Energy consumption vs. maximum team size.

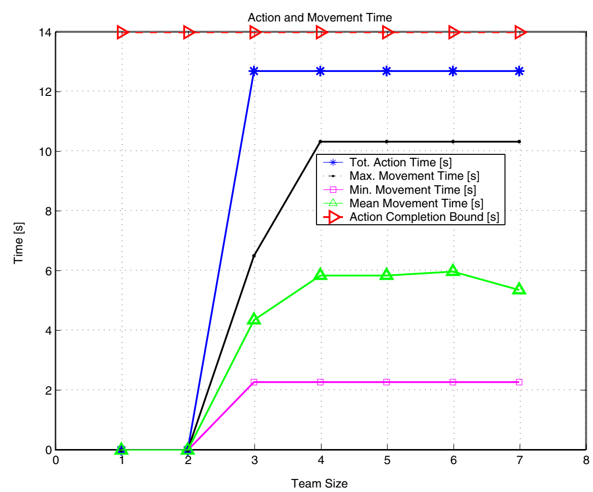

Fig. 11. Delay vs. maximum team size.

earlier drops, and fewer nodes try to transmit simultaneously. Conversely, congestion can be dramatically 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. 9), delays are two orders of magnitude lower and can be regulated with power control (Fig. 8). 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 reduced packet retransmissions at the MAC layer (Fig. 7).

B. Actor-actor Coordination

In this section, we discuss performance results for the multi-actor task allocation problem presented in Section IV. Actors are assumed to be randomly deployed in a $200\,\mathrm{m}\mathrm{x}200\,\mathrm{m}$ 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 $15\,\mathrm{s}$ and $1\,\mathrm{s}$, 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 $E_0 = 1000\,\mathrm{J}$; moreover, the velocities range in the interval $[3, 12]\,\mathrm{m/s}$.

Figures 10 and 11 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. 12 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 into an acting team is optimized

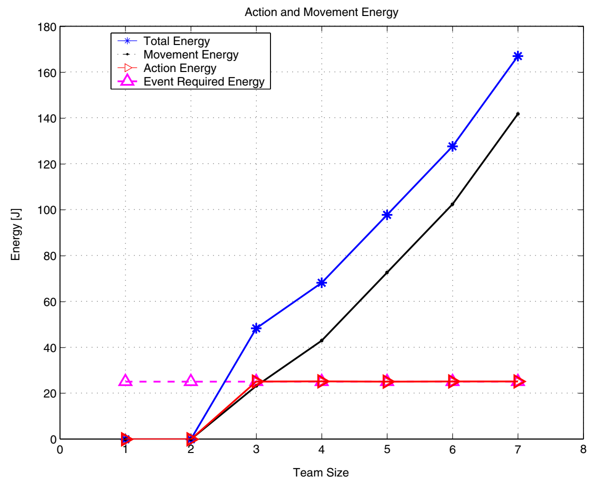

Fig. 12. Energy consumption vs. team size.

to minimize the overall energy expenditure, i.e., the sum of the movement energy E^M and the action energy E^Ω, at least 3 actors are needed to complete the action (see Fig. 10) and the total action time tends to be exactly the maximum allowed completion bound D, discounted by the coordination delay T^C (see Fig. 11). Problem $\mathbf{P}^{(\omega)}_{All}$ tends to minimize the number of involved actors, and to 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^M_{min}) 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. 12.

VI. Conclusions

We outlined the challenges for coordination and communication in Wireless Sensor and Actor Networks (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. Then, an energy efficient communication solution was derived for sensor-actor communication based on geographical routing. 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

This material is based upon work supported by the National Science Foundation under Grant No. 0428329.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks (Elsevier)*, vol. 2, no. 4, pp. 351–367, October 2004.

[2] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," in *Proceedings of IEEE INFOCOM 2005*, Miami, Fl, USA, March 2005.

[3] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1292–1302, Aug. 2004.

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

[5] M. Rahimi, H. Shah, G. Sukhatme, J. Heidemann, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipai, Taiwan, May 2003, pp. 19–24.

[6] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks," in *Proceedings of ACM Mobihoc 2005*, Urbana-Champaign, IL, May 2005.

[7] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of ACM/IEEE MobiCom 2000*, Boston, Massachussets, 2000, pp. 120–30.

[8] S. M. Das, H. Pucha, and Y. C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," in *Proceedings of IEEE INFOCOM 2005*, Miami, FL, USA, Mar. 2005.

[9] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks," in *Proceedings of IEEE SECON 2004*, Santa Clara, CA, USA, October 2004.

[10] M. Rossi and M. Zorzi, "Cost Efficient Localized Geographical Forwarding Strategies for Wireless Sensor Networks," in *Proceedings of the Tyrrhenian International Workshop on Digital Communications, TIWDC 2005*, Sorrento, Italy, July 2005.

[11] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks," *Journal of Selected Areas in Communications*, vol. 23, no. 3, pp. 520–532, Mar. 2005.

[12] F. Aurenhammer, "Voronoi Diagrams - A Survey Of A Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 23, pp. 345–405, 1991.

[13] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of IEEE INFOCOM 2001*, April 2001.

[14] P. S. Maybeck, *Stochastic models, estimation, and control - Volume 1*. New York: Academic Press, 1979.

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

[16] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering, 3rd edition*. Hoboken, NJ: John Wiley & Sons, Inc., 1996.

[17] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, pp. 51–58, May 2000.

[18] 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.

[19] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[20] The J-Sim Simulator. [Online]. Available: http://www.j-sim.org/