Communication and Coordination in Wireless Sensor and Actor Networks

Tommaso Melodia, *Student Member*, *IEEE*, Dario Pompili, *Student Member*, *IEEE*, Vehbi C. Gungor, *Student Member*, *IEEE*, and Ian F. Akyildiz, *Fellow*, *IEEE*

Abstract—In this paper, coordination and communication problems in Wireless Sensor and Actor Networks (WSANs) are jointly addressed in a unifying framework. A sensor-actor coordination model is proposed based on an *event-driven partitioning* paradigm. Sensors are partitioned into different sets, and each set is constituted by a data-delivery tree associated with a different actor. The optimal solution for the partitioning strategy is determined by mathematical programming, and a distributed solution is proposed. In addition, a new model for the actor-actor coordination problem is introduced. The actor coordination is formulated as a task assignment optimization problem for a class of coordination problems in which the area to be acted upon needs to be optimally split among different actors. An auction-based distributed solution of the problem is also presented. Performance evaluation shows how global network objectives, such as compliance with real-time constraints and minimum energy consumption, can be achieved in the proposed framework with simple interactions between sensors and actors that are suitable for large-scale networks of energy-constrained devices.

Index Terms—Wireless sensor and actor networks, mathematical programming/optimization, real-time communications, energy efficiency, task allocation.

1 Introduction

Wireless Sensor and Actor[1] Networks (WSANs) [1] are composed of heterogeneous devices referred to as *sensors* and *actors*. Sensors are low-cost low-power multifunctional devices that communicate untethered in short distances. Actors are usually resource-rich devices with higher processing capabilities, higher transmission capabilities, and longer battery life. Actors collect and process sensor data and perform actions on the environment based on the information gathered.

In WSANs, the collaborative operation of sensors enables the *distributed sensing* of a physical phenomenon. After sensors detect an event that has occurred in the environment, the event data is distributively processed and transmitted to the actors, which gather, process, and eventually reconstruct the characteristics of the event. The process of establishing data paths between sensors and actors is referred to as *sensor-actor coordination* [1]. Once the event has been detected, the actors coordinate to reconstruct it, to estimate its characteristics, and to make a collaborative decision on how to perform the action. This process is referred to as *actor-actor coordination* [1]. As a result, the operation of a WSAN can be thought of as an event-sensing, communication, decision, and acting loop.

WSANs can be seen as a distributed control system designed to timely react to sensor information with an effective action. For this reason, *real-time coordination and communication* is an important concern in WSANs to guarantee the timely execution of correct actions. The *energy efficiency* of network communications is also crucial, since sensors are resource-constrained nodes with a limited battery lifetime. Furthermore, sensor network protocols and algorithms should be *scalable* and *localized*, as the number of nodes can be arbitrarily high.

Given the above requirements, we propose basing the sensor-actor coordination on an *event-driven partitioning* paradigm in the framework of *Geographical Routing* [2]. Sensors are partitioned into different sets, and each set is constituted by a data-delivery tree associated with a different actor. The distributed partitioning is triggered by an event, and data-delivery trees are created *on-the-fly* to optimally react to the event itself and to provide the required reliability with minimum energy expenditure. In this way, only sensors in the event area are partitioned and each component of the partition consists of those sensor nodes that send their data to the same actor. Hence, event information is collected at the optimal actors, whereas existing energy resources are better utilized, since sensors are partitioned based on the localization and scope of the event and on the position of the actors. The resulting architecture is shown in Fig. 1. Our approach is inline with

1. It may be worth specifying the meaning that we attribute to the term *actor* and how this is different from the more conventional notion of *actuator*. An actuator is a device to convert an electrical control signal to a physical action and constitutes the mechanism by which an agent acts upon the physical environment. An actor, besides being able to act on the environment by means of one or several actuators, is also a *single network entity* that performs networking-related functionalities, that is, receive, transmit, and relay data. For example, a robot may interact with the physical environment by means of several motors and servomechanisms (actuators). However, from a networking perspective, it constitutes a single entity, which we refer to as actor.

- *T. Melodia is with the Department of Electrical Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260-1920. E-mail: tmelodia@eng.buffalo.edu.*
- *D. Pompili is with the Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08854-8058. E-mail: pompili@ece.rutgers.edu.*
- *V.C. Gungor is with Eaton Corporation, MI. E-mail: gungor@ece.gatech.edu.*
- *I.F. Akyildiz is with the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: ian@ece.gatech.edu.*

Manuscript received 25 Jan. 2006; revised 10 July 2006; accepted 17 Oct. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0029-0106.
Digital Object Identifier no. 10.1109/TMC.2007.1009.

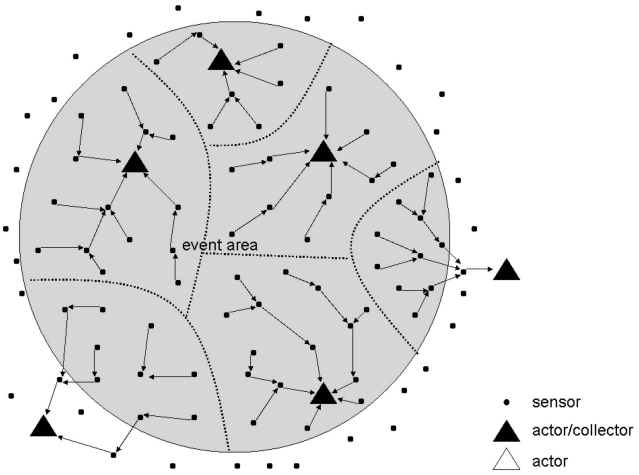


Fig. 1. Event-driven partitioning with multiple actors.

the *dynamic clustering* mechanism [3], [4]. However, the assumptions in [3] and [4] are strictly tied to the particular application considered, that is, target tracking of moving objects, whereas we consider a more general framework.

In addition, we introduce a model for actor-actor coordination whose objective is to optimally allocate tasks to the different actors to collaboratively achieve a global goal. We define an optimization model for a class of coordination problems in which the area to be acted upon is optimally split among different actors, depending on the actor's capabilities. Overall, this paper introduces a framework for communication and coordination problems in WSANs, whose contributions can be outlined as follows:

- **Sensor-actor coordination:**
  - We propose an optimization model based on the event-driven partitioning paradigm for sensor-actor coordination. This defines how sensors communicate with actors, which actor is fed by each sensor, and which data paths should be established between sensors and actors. Furthermore, we propose a new notion of *reliability* that accounts for the timely delivery of data packets at the network layer. The objective is to comply with the reliability required by the application and minimize the energy consumption. We determine the optimal strategy for event-driven partitioning by *Integer Linear Programming (ILP)* [5].
  - We propose a multistate distributed algorithm that determines sensor-actor data paths and implicitly partitions the sensors in the event area as the event occurs. The algorithm achieves an energy-efficient solution for sensor-actor coordination and is based on an adaptive mechanism that trades off energy consumption for delay when the event data must be delivered to the actors within predetermined latency bounds.
- **Actor-actor coordination:**
  - We define the actor-actor coordination problem as a task assignment problem and propose a solution for a class of coordination problems in which the area to be acted upon needs to be optimally split among different actors. The action workload is thus divided among different potentially heterogeneous actors, depending on the characteristics of the event. The task assignment problem is formulated as a *Mixed Integer Non-Linear Program (MINLP)* [6].
  - We propose a localized distributed solution for the actor-actor coordination problem based on an analogy with an auction mechanism among the actors.

Since WSANs can enable a broad range of applications with different requirements, we focus on scenarios with immobile actors that can act on a limited area defined by their *action range*[2] and the area where the event occurs needs to be monitored for a prolonged period of time. As a representative application, consider a system of simple scalar sensors that collaboratively detect the presence of an intruder. The lower tier sensors could wake up on demand pan-tilt-zoom camera/actors mounted on robotic arms that take images and video streams from the areas where the event has been detected based on sensor input. The coverage of a pan-tilt-zoom camera is defined by its field of regard, that is, the points of the physical environment that can be perceived by the camera given its ability to reposition. This would correspond to the action area of the camera. Cameras whose field of regard is overlapped would collaboratively decide on which camera-actor is best suited to gather images from the area based on the proposed model.

The remainder of this paper is organized as follows: In Section 2, we summarize the related work. In Section 3, we state the sensor-actor coordination problem and propose an ILP formulation and, in Section 4, we propose a distributed solution. In Section 5, we state the actor-actor coordination problem and, in Section 6, we introduce a distributed solution based on a real-time localized auction mechanism. Detailed comparative performance evaluation and simulation results are presented in Section 7. Finally, in Section 8, we draw the main conclusions.

2 Related Work

Although a few recent papers are specifically concerned with coordination and communication problems in sensor and actor networks, the literature on the subject is extremely limited. In [1], research challenges in WSANs are outlined and open research issues are described.

In [7], the problem of "hazards" that consist of the out-of-order execution of queries and commands resulting from a lack of coordination between sensors and actors is considered. Three types of hazards are identified, and their undesirable consequences are shown. The authors also identify and enumerate the associated challenges in addressing hazards and propose a distributed hazard-free approach that addresses the problem and its challenges. The problem of avoiding hazards resulting from the out-of-order execution of queries is of great importance in WSANs and is complementary to our work.

Some recent papers [8], [9] have considered the issue of real-time communication in sensor networks. The SPEED protocol [8] provides real-time communication services and is designed to be a stateless localized algorithm with low

2. Actors are immobile from a networking perspective but may contain moving mechanical parts. The notion of action range may refer, for example, to the extension of mechanical arms that perform an action or to the range of automatic water sprinklers.

control overhead. End-to-end soft real-time communication is achieved by maintaining the desired delivery speed across the sensor network through a combination of feedback control and nondeterministic geographic forwarding. MMSPEED [9] 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, that is, defining how actors and sensors coordinate and communicate, or with actor-actor coordination.

Several solutions propose to guarantee scalability and energy efficiency based on partitioning the sensor network into different clusters [10], [11], [12]. Most of the existing clustering algorithms can be classified as *topology-dependent*; that is, clusters are predetermined, depend on the topology of the sensor network, and may be adaptively reconfigured to deal with the mobility or failure of the sensor nodes. Conversely, similar to the dynamic clustering approach in [3], in the event-driven partitioning, paradigm sensors are partitioned based on the characteristics of the event.

3 Sensor-Actor Coordination: Problem Formulation

As discussed in Section 2, sensor-actor communications may have real-time requirements. Hence, we introduce a novel notion of reliability that accounts for the percentage of the packets generated by the sensors in the event area and received within a predefined latency bound. Unlike other notions of reliability, the definition introduced here is related to the real-time delivery of data packets from sources to actors and is calculated at the network layer.

Definition 1. *The* latency bound B *is the maximum allowed time between the instant when the physical features of the event are sampled by the sensors and the instant when the actor receives a data packet describing these event features.*

Definition 2. *A data packet that does not meet the latency bound B when it is received by an actor is said to be* expired *and, thus,* unreliable. *Similarly, a data packet received within the latency bound B is said to be* unexpired *and, thus,* reliable.

Definition 3. *The* event reliability r *is the ratio of* reliable *data packets over all the packets generated in a decision interval.*[3] *The* event reliability threshold r_{th} *is the minimum* event reliability *required by the application.*

Definition 4. *The* lack of reliability *is the difference $(r_{th} - r)$ between the required event reliability threshold r_{th} and the observed event reliability r at a given time.*

Note that the latency bound B and the event reliability threshold r_{th} are dependent on the application requirements.

The sensor-actor coordination problem consists of establishing data paths from each sensor residing in the event area to the actors by 1) ensuring that the observed reliability r is above the event reliability threshold r_{th} (that is, $r \geq r_{th}$) and 2) minimizing the energy consumption associated with data-delivery paths.

3. When one or more packets are dropped, the actor is notified about the lost packet(s) in the header of the next data packet to account for the lost packet in the measured reliability.

We refer to our solution for the sensor-actor coordination problem as *event-driven partitioning with multiple actors* and model it as an ILP. In Section 3.1, we describe the network and energy model. In Section 3.2, we provide the complete ILP formulation of the problem.

3.1 Network and Energy Model

The network of sensors and actors is represented as a graph $\mathcal{G}(\mathcal{S}^{\mathcal{V}}, \mathcal{S}^{\mathcal{E}})$, where $\mathcal{S}^{\mathcal{V}} = \{v_1, v_2, \ldots, v_N\}$ is a finite set of nodes (vertices) in a finite-dimension terrain, with $N = |\mathcal{S}^{\mathcal{V}}|$ and $\mathcal{S}^{\mathcal{E}}$ is the set of links (edges) among nodes, that is, $e_{ij} \in \mathcal{S}^{\mathcal{E}}$ iff nodes v_i and v_j (also, i and j for simplicity in the following) are within each other's transmission range. Let $\mathcal{S}^{\mathcal{A}}$ represent the set of actors with $N_A = |\mathcal{S}^{\mathcal{A}}|$. We refer to an actor that is collecting traffic from one or more sources as a *collector*. Let $\mathcal{S}^{\mathcal{S}}$ be the set of traffic sources, with $N_S = |\mathcal{S}^{\mathcal{S}}|$. This set represents the sensor nodes that detect the event, that is, the sensors that reside in the event area. Since the set of sources is disjoint from the set of actors $\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}^{\mathcal{V}}$, $\mathcal{S}^{\mathcal{S}} \subset \mathcal{S}^{\mathcal{V}}$, and $\mathcal{S}^{\mathcal{A}} \cap \mathcal{S}^{\mathcal{S}} = \emptyset$, we define $\mathcal{P} = \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}\}$ as the set of source-destination connections.

An accurate model for the energy consumption per bit at the physical layer is $E = E_{elec}^{trans} + \beta d^{\alpha} + E_{elec}^{rec}$, where E_{elec}^{trans} is a *distance-independent* term that takes into account overheads of transmitter electronics (phase-locked loops, voltage-controlled oscillators, bias currents, and so forth) and digital processing, E_{elec}^{rec} is a *distance-independent* term that takes into account the overhead of receiver electronics, and βd^{α} is a *distance-dependent* term that accounts for the radiated power necessary to transmit one bit over a distance d between the source and the destination. As in [10], we assume that $E_{elec}^{trans} = E_{elec}^{rec} = E_{elec}$. Thus, the overall expression simplifies to $E = 2E_{elec} + \beta d^{\alpha}$, where α is the exponent of the path loss $(2 \leq \alpha \leq 5)$, β is a constant $[\mathrm{J}/(\mathrm{bit} \cdot \mathrm{m}^{\alpha})]$, and E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit $[\mathrm{J/bit}]$. We assume that, when a sensor node receives data from at least two other nodes, it can aggregate the received information by *data fusion* [13], that is, by merging multiple incoming packets and, thus, reducing the amount of data to be transmitted. To effectively support this function, an algorithm for data fusion should be implemented on each sensor, which is out of the scope of this paper. Moreover, we ignore the processing cost, since it is much lower than the communication cost [14].

3.2 Integer Linear Problem

The objective of the optimization problem is to find *data aggregation trees* (da-trees) from all the sensors that reside in the event area (referred to as sources) to the appropriate actors. A da-tree is composed by aggregating individual *flows*, where a flow is defined as a connection between a sensor and an actor. All leaves in a da-tree are sources (but not all sources are necessarily leaves), and each actor is either the root of a da-tree or does not participate in the communication. The da-trees are constructed in such a way that each source belongs to one tree only, and each tree has only one actor as its root. Event-driven partitioning can thus be seen as a twofold problem: 1) select the optimal subset of actors to which sensor readings will be transmitted and 2) construct the minimum energy da-trees toward those selected actors that meet the required event reliability constraint. The union of all trees rooted at the actors

implicitly partitions the set of source nodes in the event area. Fig. 1 shows an example of this configuration.

The event-driven partitioning problem is formulated as an ILP [5]. The network topology is assumed to be *1-connected*; that is, at least one path exists between each sensor and actor. We introduce the following notation:

- e_{ij} is a binary variable that represents a link and equals 1 iff nodes i and j are within each other's transmission range.
- c_{ij} is the cost of the link between nodes i and j, that is, $2E_{elec} + \beta d_{ij}^{\alpha}$, where d_{ij} is the distance between nodes i and j.
- x_{ij}^{k} is a binary variable equal to 1 iff link (i,j) is part of the da-tree associated with actor k.
- $f_{ij}^{k,s}$ is a binary variable equal to 1 iff source s sends data to actor k and link (i,j) is in the path from s to k.
- $l^{k,s}$ is a binary variable equal to 1 iff sensor s sends data to actor k.
- p_{ij} is the propagation delay associated with link (i,j), defined as d_{ij}/v, where v is the signal propagation speed.
- $\tilde{d}$ is a parameter that accounts for *processing*, *queuing*, and *medium access* delay at each sensor node.
- B is the latency bound on each source-actor flow.
- r and r_{th} are the event reliability and the required event reliability threshold, respectively.
- $b^{k,s}$ is a binary variable equal to 1 iff the connection between source s and actor k is not compliant with the latency bound; that is, the end-to-end delay is higher than the latency bound B.
- Q is the number of noncompliant sources.

The problem can be cast as follows:

$\mathbf{P}_{Min}^{Com}$: Event-Driven Partitioning with Multiple Actors

$$Given: \quad e_{ij},\ c_{ij},\ p_{ij},\ v,\ \tilde{d},\ B,\ r_{th}$$

$$Find: \quad x_{ij}^{k},\ f_{ij}^{k,s},\ l^{k,s},\ b^{k,s},\ r$$

$$Minimize: \quad C^{TOT} = \sum\nolimits_{k\in\mathcal{S}^A} \sum\nolimits_{(i,j)\in\mathcal{S}^E} x_{ij}^{k}\cdot c_{ij} + \gamma\cdot Q \tag{1}$$

$Subject\ to:$

$$\sum_{j\in\mathcal{S}^V}(f_{sj}^{k,s} - f_{js}^{k,s}) = l^{k,s}, \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \tag{2}$$

$$\sum_{j\in\mathcal{S}^V}(f_{kj}^{k,s} - f x_{jk}^{k,s}) = -l^{k,s}, \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \tag{3}$$

$$\sum_{j\in\mathcal{S}^V}(f_{ij}^{k,s} - f_{ji}^{k,s}) = 0,$$

$$\forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \forall i\in\mathcal{S}^V\ s.t.\ i\neq s, i\neq k, \tag{4}$$

$$f_{ij}^{k,s} \le e_{ij}, \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \forall i\in\mathcal{S}^V, \forall j\in\mathcal{S}^V, \tag{5}$$

$$f_{ij}^{k,s} \le x_{ij}^{k}, \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \forall i\in\mathcal{S}^V, \forall j\in\mathcal{S}^V, \tag{6}$$

$$\sum_{k\in\mathcal{S}^A} l^{k,s} = 1, \forall s\in\mathcal{S}^S, \tag{7}$$

$$f_{ij}^{k,s} \le l^{k,s}, \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \forall i\in\mathcal{S}^V, \forall j\in\mathcal{S}^V, \tag{8}$$

$$\varepsilon\cdot[B - \sum_{(i,j)\in\mathcal{S}^E} f_{ij}^{k,s}(p_{ij}+\tilde{d})] \le b^{k,s}, \quad \forall s\in\mathcal{S}^S, \forall k\in\mathcal{S}^A, \tag{9}$$

$$Q = \sum_{k\in\mathcal{S}^A}\sum_{s\in\mathcal{S}^S} b^{k,s}; \qquad r = \frac{|\mathcal{S}^S| - Q}{|\mathcal{S}^S|} \ge r_{th}. \tag{10}$$

The objective function in (1) minimizes the overall energy consumption and imposes a penalty by multiplying the number of noncompliant sources Q by a penalty coefficient γ whose value must be high enough (for example, orders of magnitude higher than the energy consumption) to guarantee the uniqueness of the solution. This allows for minimizing the number of noncompliant sources Q in (10) with a single-objective problem. As previously discussed, a flow is a connection between a source and an actor. Flows associated with the same actor are aggregated in a da-tree. Constraints (2), (3), and (4) express the conservation of flows [5]; that is, each source generates a flow, which is collected by an actor. Constraint (5) ensures that flows are created on links between adjacent nodes (that is, those within the transmission range of each other). Constraint (6) forces all flows from different sources but directed toward the same actor to be aggregated in the tree associated with that actor. Constraint (7) imposes that each source send data to exactly one actor. Constraint (8) ensures that all flow variables from a source to a particular actor are 0 unless that actor is selected by the source. Constraint (9) requires that the binary variable $b^{k,s}$ be equal to 1 if and only if the flow between source s and actor k violates the latency bound B. The small negative coefficient ε scales the value in brackets to make it smaller than 1. Hence, when the latency bound is violated, the left side of (9) is a small positive value, which forces the binary variable $b^{k,s}$ to be 1. Conversely, when the latency bound is met, the left side of (9) is negative and $b^{k,s}$ will assume the 0 value to minimize the objective function in (1). Finally, in (10), Q is defined as the number of noncompliant sources, and the reliability r is calculated as the ratio of compliant sources over all sources and is constrained to be over the required threshold.

Since $\mathbf{P}_{\mathbf{Min}}^{\mathbf{Com}}$ is an ILP, it can be shown that it is at least as complex as the Geometric Connected Dominating Set problem, which is proven to be NP-complete [15]. Hence, $\mathbf{P}_{Min}^{Com}$ is NP-complete. However, it is still possible to solve $\mathbf{P}_{Min}^{Com}$ for networks of moderate size (up to 100 nodes) as will be shown in Section 7. This allows for gaining an insight into the properties of the optimal solution and designing distributed solutions that try to reproduce characteristics of the optimal network configuration. The design of the distributed protocol presented in Section 4 is based on the analysis and performance study of the above problem. Moreover, the mathematical formulation constitutes a fundamental benchmark to evaluate the performance of distributed solutions. In this spirit, in Section 7.1, we will use it as a benchmark for the performance of the distributed suboptimal but scalable algorithm introduced in Section 4.

4 Sensor-Actor Coordination: Distributed Protocol

The objective of the distributed protocol proposed in this section is to build da-trees between the sources that reside in the event area and the actors in such a way as to minimize the objective function in (1), that is, to provide the required reliability r_{th} with minimum energy expenditure. As will be shown in Section 7.1, the proposed protocol constructs da-trees between sources and actors that can

be seen as an approximate solution to the event-driven partitioning with the multiple-actors problem described in Section 3. We refer to the protocol as *Distributed Event-driven Partitioning and Routing* (DEPR) protocol.

DEPR relies on local information and on *greedy* routing decisions, thus resulting in a good compromise between the energy efficiency of the da-trees and the amount of topology information needed by each sensor to take a routing decision [2]. Conversely, complying with predetermined delay bounds requires some form of end-to-end feedback. Instead of requiring feedback information for each individual source, which would cause an unacceptable overhead, we rely on collective feedback from the receiving actors as will be explained in Section 4.5. Each actor advertises the observed reliability. Based on this, the proposed protocol favors the local behavior for each individual sensor node that results in a global network behavior that is compliant with the application requirements, that is, provide an event reliability r above the required threshold r_{th} (Definition 3 in Section 3) and minimize the energy consumption. The reliability is controlled based on the idea of adjusting the delays by modifying the average end-to-end path length. Although modifying the energy consumption in an ad hoc network by changing the transmitted power is a common practice, the proposed protocol can also be seen as a mechanism to adjust the end-to-end delay based on transmission power control. To the best of our knowledge, this idea has not been thoroughly explored so far.

In the description of the DEPR protocol, we assume that 1) each sensor is aware of its position, 2) each sensor is aware of the position of its neighbors and of the actors, and 3) the network is synchronized by means of one of the existing synchronization protocols [16].

An important issue in geographical routing algorithms is to avoid routing loops. Hence, we introduce some concepts related to *path loop freedom*.

Definition 5. *Given nodes v and x, the* absolute advance *of node x with respect to v is the distance between v and its closest actor c_v minus the distance between x and its closest actor c_x.*[4]

Definition 6. *Given nodes v and x, the* advance toward the collector *c of x with respect to v is the distance between v and c minus the distance between x and c.*

Intuitively, if x has a *positive absolute advance* with respect to v, then x is closer to one actor than v. If x has a *positive advance toward collector c* with respect to v, then x is closer to actor c than v. For any multihop path, a *positive absolute advance* at every hop guarantees loop freedom irrespective of the final destination, since, at each hop, the packet is closer to a collector than at the previous hop. A *positive advance toward an actor c* at every hop guarantees a loop-free path from a source node to the actor c.

4.1 Overview of DEPR

Each sensor alternates among four different states, namely, *idle*, *start-up*, *speed-up*, and *aggregation states*, plus an additional *recovery state* that will be discussed in Section 4.6. An overview of the state transitions is depicted in Fig. 2. The main objective of the state transitions is to reduce the number of hops, which results in a decreased delay when the reliability requirement is violated and to save energy when the reliability requirement is met. This is achieved by probabilistically modifying the behavior of the sensor nodes at the routing layer and physical layer, that is, inducing them to select their next hop so as to increase the delay and reduce the energy consumption when the reliability is high, and vice versa, reducing the delay at the expense of energy consumption when the reliability is low. This is achieved by dynamically adjusting the transmit power at the same time.

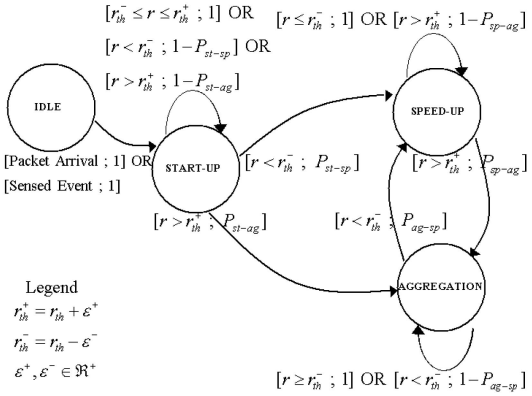


Fig. 2. State-transition diagram for a sensor node.

For each decision interval, each actor computes the event reliability r as the ratio of unexpired packets over all generated packets and periodically broadcasts its value. Sensors associated with that collector base their state transitions on the reliability observed by the collector, which is broadcast at the end of each decision interval. When the advertised value r is below the so-called *low event reliability threshold* r_{th}^-, where $r_{th}^- = r_{th} - \epsilon^-$, that is, the lack of reliability $(r_{th} - r)$ is above a certain positive margin ϵ^-, it is necessary to speed up the data-delivery process by reducing the end-to-end delay. Conversely, when the advertised value r is above the so-called *high event reliability threshold* r_{th}^+, where $r_{th}^+ = r_{th} + \epsilon^+$, there is reliability in excess that can be traded off for energy savings. The parameters ϵ^+ and ϵ^- are needed to define a "tolerance zone" around the required reliability threshold for practical purposes (that is, reduce oscillations).

Each sensor node starts in the idle state, where it samples the environment and monitors the channel for incoming data packets. A sensor enters the start-up state when it either senses an event or receives the first data packet from a neighboring sensor. The collective operation of the sensor nodes in the start-up state allows for establishing paths to an actor for each source that resides in the event area. These paths constitute a good compromise between latency and energy consumption.

Sensor nodes then wait for feedback messages from the collector/actor that they are associated with. If the event reliability r is advertised to be below the low event reliability threshold r_{th}^-, then it is necessary to reduce the sensor-actor delay by reducing the end-to-end path length. Hence, when it receives a packet advertising a reliability below the low reliability threshold $(r < r_{th}^-)$, a sensor in the start-up state enters the speed-up state with probability P_{st-sp}. This is further discussed in Section 7.2. The notation

4. Note that c_v and c_x can be different actors.

$[cond; P]$ in Fig. 2 indicates a transition that occurs with probability P when the condition $cond$ is verified.

If the event reliability r is above the high event reliability threshold r_{th}^{+} (that is, $r > r_{th}^{+}$), then it is possible to save energy. In this case, a node in the start-up state enters the aggregation state with probability P_{st-ag}, where it tries to minimize the energy consumption associated with its transmission by relaying data to the closest neighbor that participates in a da-tree.

Then, sensors alternate between the speed-up and aggregation states to respond to feedback messages from collectors. Hence, as shown in Fig. 2, a sensor in the speed-up state enters the aggregation state, with probability P_{sp-ag}, when $r > r_{th}^{+}$, whereas a sensor in the aggregation state enters the speed-up state, with probability P_{ag-sp}, when $r < r_{th}^{-}$. The objective of the protocol is to converge to a solution with a reliability close to the event reliability threshold with minimal energy consumption. A sensor goes back to the idle state if it does not generate or receive packets for $idleTimeout$ seconds.

In Sections 4.2, 4.3, and 4.4, we describe the operations of each state.

4.2 Start-Up State

As shown in Fig. 2, a node enters the start-up state from the idle state when it detects an event or when it receives a packet to be relayed to an actor. Sensor i in the start-up state, either as a source or as a relayer for a data packet, selects the next hop based on the so-called *two-hop rule*. According to the two-hop rule, node i selects node j as the next hop, which minimizes the sum of energy consumption from i to j and the energy consumption from j to the actor closest to j (c_j), which is computed according to the link energy model introduced in Section 3.1. Hence, the energy consumption E_j associated with a neighbor j of i is

$$E_j = 2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{jc_j}^{\alpha}, \tag{11}$$

where d_{ij} represents the distance between i and j and d_{jc_j} represents the distance between j and its closest actor c_j. Note that the latter link may or may not exist. The two-hop rule selects the node j associated with the minimum two-hop energy consumption as the next hop. As a result, the source-actor path will be established by applying the two-hop rule iteratively. Note that this procedure is only based on local position information. It requires each node to know only the position of its neighbors and of the actors and does not entail any other exchange of information. The operations executed by a sensor node in the start-up state are detailed in Algorithm 1.

Algorithm 1 Start-Up State
Pseudocode executed by node v_i in the start-up state
$mincost = \infty$
if (($I\_am\_a\_source$) **or** ($I\_am\_a\_relayer$)) **then**
  for each of my neighbors v_j **do**
    for each actor s_k **do**
      if $(2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{js_k}^{\alpha}) < mincost$ **then**
        $mincost = 2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{js_k}^{\alpha}$
        $nexthop = v_j$
      end if
    end for
  end for
end if
Inform $nexthop$ that it is a relayer

The two-hop rule produces loop-free paths, as stated below.

Lemma 1. *The next hop j selected by a node i with the* two-hop rule *has a* positive absolute advance *with respect to i (see Definition 5).*

Proof. Assume that node s_1 is holding a message to be relayed to an actor, either a_1 or a_2. Let us also assume, without loss of generality, that the two-hop path $s_1 - s_2 - a_1$ is more energy efficient than the direct link $s_1 - a_1$, that is, $4E_{elec} + \beta(d_1^{\alpha} + d_2^{\alpha}) \leq 2E_{elec} + \beta d^{\alpha}$, which leads to $(d_1^{\alpha} + d_2^{\alpha}) < d^{\alpha}$. Let us now assume that a_1 is the closest actor to node s_1 and that s_3 is the best next hop according to the two-hop rule; that is, the energy necessary to reach a_2 through s_3 is lower than the energy required to reach a_1 through s_2, according to the energy metric in Section 3.1. This directly translates into $d_3^{\alpha} + d_4^{\alpha} < d_1^{\alpha} + d_2^{\alpha}$. Hence, we have $d_3^{\alpha} + d_4^{\alpha} < d_1^{\alpha} + d_2^{\alpha} < d^{\alpha}$, which ultimately means, being $\alpha \geq 2$, that $d > d_4$. Therefore, the energy-efficient next hop always has a positive absolute advance. □

As a consequence, applying the two-hop rule at each hop produces a loop-free path between the source and the actor.

4.3 Speedup State

The objective of the speed-up state is to minimize the number of hops between sources and actors. This is achieved by applying the Greedy Routing Scheme (GRS) [17] forwarding rule. According to GRS, each node sends the packet to the node closest to the destination within the transmission range. It is intuitive that this rule minimizes the number of hops in the path, the distance traveled by the packet, and the number of transmissions of the same data packet. The pseudocode of the operations executed by a sensor node in the speed-up state is reported in Algorithm 2. The set P_i in the algorithm represents the subset of the neighbors of v_i with absolute positive advance with respect to v_i.

Algorithm 2 Speedup State
Pseudocode executed by node v_i in the speed-up state
for each node $v_j \in P_i$ **do**
  if $(\text{distance}(v_i, v_j) > \text{distance}(v_i, next\_hop))$ **then**
    $next\_hop = v_j$
  end if
end for

4.4 Aggregation State

The objective of the aggregation state is to reduce the overall energy consumption. To this end, sensor nodes in the aggregation state take routing decisions that reduce the global energy consumption by relying on the data fusion algorithm that we assume to be implemented on each sensor. Since data packets can be aggregated by any node in the network, the objective of a node in the aggregation state is to route data to the closest node in its neighborhood that is part of a da-tree.

As previously discussed, after da-trees are established, each sensor knows which collector-actor it is associated with. By overhearing transmissions on the shared medium, each sensor learns which da-tree its neighbors are associated with. Hence, node v_i in the aggregation state first evaluates the cost of transmitting data to those among its neighbors that are part of a da-tree. We emphasize that this does not incur any overhead other than for overhearing packets.

Two different situations can occur. Node v_{min} can be either on the same da-tree as v_i and, hence, associated with the same collector, or in a different da-tree. If v_{min} is in the same da-tree as v_i, then v_{min} can be selected as next hop by v_i only if it has a positive advance toward the collector that both nodes are associated with, that is, if v_{min} is closer than v_i to the collector (see Definition 6). This guarantees loop freedom. In the resulting da-tree, any parent node is guaranteed to have a positive advance toward the collector with respect to each child. When v_{min} is selected, the individual transmission cost for v_i is locally minimized and the overall cost of the tree is thus reduced.

Another possible situation occurs when v_{min} is associated with a different collector other than v_i; that is, v_i and v_{min} are in two different da-trees. In this case, v_i is allowed to select v_{min} as its next hop only if v_i is a leaf in its da-tree and v_{min} has a positive advance toward its actor with respect to v_i. This guarantees loop-freedom of the overall tree, as every parent node is assured to have a positive advance toward the actor with respect to each child. Conversely, it can be easily shown that, if nonleaf nodes are allowed to switch from one da-tree to another, then loops may be created, as the condition that every parent node is closer to the actor than each child does not necessarily hold. The detailed operations executed by a sensor node in the aggregation state are given in Algorithm 3.

Algorithm 3 Aggregation State

Pseudocode executed by a node v_i in the aggregation state

for each of my active neighbors v_j **do**
  if $(\text{distance}(v_i, v_j) < \text{distance}(v_i, nexthop))$ **then**
    $v_{min} = v_j$
  end if
end for
$s = \text{actor}(v_{min})$
if $(s == myactor)$ **then**
  if $\text{distance}(v_{min}, s) < \text{distance}(v_i, s)$ **then**
    $nexthop = v_{min}$
  else
    delete v_{min} from list and restart Aggregation State
  end if
else if $I\_am\_a\_leaf$ **then**
  $nexthop = v_{min}$
else
  delete v_{min} from list and restart Aggregation State
end if

4.5 State Transitions

The transition of sensor nodes among states is driven by feedback messages from the actors. Hence, the proposed mechanism can be seen as a form of closed-loop control at the network layer. Feedback messages are periodically sent by each actor, with a period equal to Δf seconds. At each decision instant k, the actor feedback is determined based on three different reliability measures, namely, the *reliability* $r[k]$, the *short-term reliability* $r_{sh}[k]$, and the *predicted reliability* $\hat{r}[k+1]$. The actor calculates the reliability $r[k]$ observed during the last decision interval, whose length is Δd, as discussed in Section 3. Similarly, it calculates the so-called short-term reliability $r_{sh}[k]$ as the reliability observed during the last short decision interval of length Δd_{sh}, with $\Delta d_{sh} < \Delta d$. Based on the current reliability $r[k]$ and on the history of past measurements $r[k-1], r[k-2], \ldots$, the actor calculates the predicted reliability $\hat{r}[k+1] = f(r[k], r[k-1], r[k-2], \ldots)$. The feedback packet contains the advertised value of reliability $r_{adv}[k]$, calculated on the basis of these three measures, and is actually sent only if the advertised value of reliability is above the high reliability threshold r_{th}^{+} or below the low reliability threshold r_{th}^{-}. If sensors receive no feedback, then they assume the reliability to be within r_{th}^{+} and r_{th}^{-}.

The operation of the adaptive control scheme run at each actor is summarized in Algorithm 4. The advertised reliability $r_{adv}[k]$ is calculated as follows: As a general rule, the advertised value $r_{adv}[k]$ at instant k is the predicted reliability $\hat{r}[k+1]$. In this way, the actor tries to identify ongoing trends in the value of reliability and react accordingly. However, a series of conservative countermeasures is taken to minimize the probability that the reliability drops below the low threshold r_{th}^{-}, which constitute *exceptions* to the general rule. Hence, no feedback is sent when the values of reliability $r[k]$ and short-term reliability $r_{sh}[k]$ are within the two thresholds, even if the value of the predicted reliability $\hat{r}[k+1]$ is above the high threshold r_{th}^{+} (Exception 1 in Algorithm 4). Furthermore, when the value of the predicted reliability $\hat{r}[k+1]$ is within r_{th}^{+} and r_{th}^{-} or above r_{th}^{+} and when the value of the short-term reliability $r_{sh}[k]$ is within the thresholds, but the actual value of the reliability $r[k]$ is below r_{th}^{-}, an uptrend in the reliability is identified, which needs to be consolidated and accelerated by sending a feedback that advertises low reliability $r[k]$ (Exception 2 in Algorithm 4). Finally, whenever the value of the short-term reliability $r_{sh}[k]$ drops below the threshold r_{th}^{-}, the advertised value $r_{adv}[k]$ is set equal to the short-term reliability $r_{sh}[k]$ irrespective of the value of the reliability $r[k]$ and of the predicted reliability $\hat{r}[k+1]$ (Exception 3 in Algorithm 4). This is done to preemptively invert a downtrend before the reliability actually drops below r_{th}^{-}. The rule defined in Exception 3 has priority over all the other rules.

Algorithm 4 Adaptive Control Scheme

Pseudocode executed by each actor

$r_{adv}[k] = \hat{r}[k+1]$
`// General Case`
if $(\text{isAbove}(\hat{r}[k+1])$ **or** $\text{isBelow}(\hat{r}[k+1]))$ **then**
  $\text{feedbackNeeded} = \textbf{true}$
end if
`// Exception 1: Slow uptrend`
if $(\text{isAbove}(\hat{r}[k+1])$ **and** $\text{isWithin}(r[k])$ **and**
$\text{isWithin}(r_{sh}[k]))$ **then**
  $\text{feedbackNeeded} = \textbf{false}$

end if
// Exception 2: Consolidate uptrend
if (isBelow($r[k]$) **and** isWithin($r_{sh}[k]$) **and**
!isBelow($\hat{r}[k+1]$)) **then**
  feedbackNeeded = **true**
  $r_{adv}[k] = r[k]$
end if
// Exception 3: Low short-term reliability
if (isBelow($r_{sh}[k]$)) **then**
  feedbackNeeded = **true**
  $r_{adv}[k] = r_{sh}[k]$
end if
if (feedbackNeeded) **then**
  sendFeedback($r_{adv}[k]$)
end if

4.6 Handling Voids

In geographical routing protocols, nodes can work in either a *greedy mode* or a *recovery mode*. When in the greedy mode, the node that currently holds the message tries to forward it toward the destination. The recovery mode is entered when a node fails to forward a message in the greedy mode, since none of its neighbors has a positive advance toward the destination. Usually, this occurs when the node observes a *void region* between itself and the destination. Such a node is referred to as *concave* node. A packet enters the recovery routing mode when it reaches a concave node and resumes greedy forwarding when it reaches a node that is closer to the destination than the concave node. Several schemes [18], [19] that are based on *face routing* on *planar graphs* have been proposed to solve this problem. The main drawback of these solutions is that they may select long detouring paths [20]. When a packet reaches a concave node, recovery routing algorithms select a left or right detour path according to predefined rules. As a result, they may select long detouring paths. For example, the hop count of detouring paths constructed by FACE-2 [19] is, on the average, twice that of the shortest detouring path and with a much higher variance [20].

We combine face routing, in particular, the FACE-2 algorithm, with our distributed algorithm. The objective is twofold: 1) the detouring path needs to be based on paths constructed by FACE-2, thus guaranteeing delivery, and 2) the path length still needs to be adjusted based on the reliability observed at the actor.

The operations of the recovery mode are as follows: Assume that sensor v generates or receives a packet and v identifies itself as a concave node in the path toward its closest actor c_v. All neighbors with an *absolute positive advance* with respect to v are feasible next hops. This includes all neighbors w whose distance to their closest actor c_w is smaller than the distance between v and its closest actor c_v. If no such neighbor exists, v resorts to transmitting the packet toward its closest actor c_v through a detouring path and thus enters the recovery mode. To accomplish this, v enters the *recovery state*, where the next hop is selected according to the rules defined in FACE-2. In the recovery state, nodes transmit at their maximum power to allow all neighboring nodes to overhear their transmissions. The packets transmitted by nodes in the recovery state contain in their header a detouring hop number that identifies their position in the detouring path and which we refer to as their *virtual proximity* to the destination actor. Hence, the packet transmitted by the first concave node in the path will have virtual proximity 1, and so on, increasing toward the actor. When in recovery state, a generic node v initially ignores feedback messages from the actor. At the same time, v listens to the channel for packets transmitted by neighboring nodes. Whenever a node v on the detouring path overhears a packet transmitted by a neighbor w destined to the same actor and with a higher virtual proximity than its own, v flags w as a feasible next hop since it is part of the detouring path toward c_v and has a positive virtual advance (that is, w has a virtual proximity higher than v). Through overhearing, v thus constructs the list of neighbors that are part of the detouring path toward c_v, along with their virtual proximity. Then, the operation resumes according to the rules given in Section 4. Only, decisions are now based on virtual proximity, that is, the relative position on the detouring path. The speed-up state selects the most advanced neighbor in the detouring path instead of the neighbor that is physically closest to the actor, whereas the aggregation state selects the closest neighbor in the detouring path with a positive virtual advance.

5 Actor-Actor Coordination: Problem Formulation

The objective of the actor-actor coordination process is to select the best actor(s) to perform an action on the event area. At the end of the sensor-actor coordination phase described in Section 4, one or multiple actors, which we denote as *collectors*, receive sensor readings from source sensors that define the *event area*. The event area corresponds to the *action area*, that is, the area where an action is required. In particular, each collector receives data from a subset of the sources. Each element in the partition in Section 3 identifies a portion of the action/event area and is under the responsibility of the corresponding collector. However, the collector may not be able to act on the entire area that it is responsible for, since the area may not be totally within the collector's *action range*. The action range defines the circular area where an actor is able to act. Moreover, the collector may not be the "best" actor for that task in terms of *action completion time* and/or *energy consumption*, where the former is the time to perform the action and the latter is the required energy for the action. For these reasons, an actor-actor coordination is required before initiating the action.

Definition 7. *The* action completion bound *is the maximum allowed time from the instant when the event is sensed to the instant when the action is completed.*

The coordination objective of each collector-actor is to find the optimal actors to timely act on the portion of the event area under its own responsibility. In particular, if multiple actors can act on a certain area, then we refer to the area as an *overlapping area* (region areas numbered from 1 to 8 in Fig. 3). In an overlapping area, the actor-actor coordination problem consists of selecting a subset of the actors and their action powers to optimally divide the action workload so as to maximize the *residual energy* to

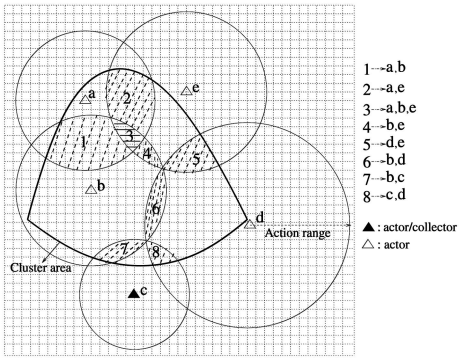


Fig. 3. Overlapping and nonoverlapping areas for collector c.

extend the lifetime of the actors[5] while complying with the *action completion bound*. We refer to an area where only one actor can act as a *nonoverlapping area* (unshaded regions in Fig. 3). For such an area, the coordination problem simplifies to selecting the power level for the actor that minimizes the energy consumption while abiding by the action completion bound. For this reason, we assume that the coordination problem involves only overlapping areas and that the available energy of each actor is already discounted with the energy needed to act on nonoverlapping areas.

Let $\mathcal{S}^A$ represent the set of actors, with $N_A = |\mathcal{S}^A|$, and let $\mathcal{S}^C$ be the set of collectors ($\mathcal{S}^C \subseteq \mathcal{S}^A$). As mentioned before, collectors receive data from sources (sensors) and, from the source positions, they can identify the portion of the whole event area that they are responsible for. By referring to Fig. 3, we introduce the following notation:

- $\mathcal{A}^h_{c,nov}$ and $\mathcal{A}^m_{c,ov}$ are the hth *nonoverlapping* and the mth *overlapping* areas, respectively, inside the portion of the event area under the responsibility of collector c. H_c represents the number of nonoverlapping areas, whereas M_c represents the number of overlapping areas associated with collector c.
- $\mathcal{S}^{A,m}_{c,ov}$ is the set of actors that can act on the mth overlapping area $\mathcal{A}^m_{c,ov}$ that is under the responsibility of collector c.

Each actor a is characterized by the following parameters:

- $R_a\ [m]$ is the action range of a.
- $P^{Max}_a\ [W]$ is the maximum power that actor a can use to perform the action. Actors can select their power among L different levels

$$P_{a,p} = \frac{P^{Max}_a}{L} \cdot p, p = 1, 2, \ldots, L, \tag{12}$$

  where $P_{a,p}$ is the pth power level for actor a. As will be shown in (13), a higher power corresponds to a lower action completion time.
- η_a is the *efficiency* of actor a (see (13)).
- $E^{Av}_a[J]$ is the available energy of actor a, discounted with the energy needed to act on nonoverlapping areas where only actor a can act.

5. Although actors are resource-rich nodes, the order of magnitude of the energy required for actions is higher than that required for communication. Hence, it is important to save action energy to extend the lifetime of actors.

We formulate the actor-actor coordination problem as MINLP. The objective is to find, for each portion of the event area, the subset of actors that maximizes the average residual energy of the actors involved in the action under the constraint of meeting the action completion bound. It is assumed that the energy required to perform the action is orders of magnitude higher than the energy required for communication.

Let us introduce the following notation:

- $P^{(m)}_{a,p}\ [W]$ is the pth power level of actor a for the mth overlapping area $\mathcal{A}^m_{c,ov}$, whose measure is $A^m_{c,ov}\ [m^2]$.
- $\underline{X}^{(m)}$ is a binary matrix whose element $[x^{(m)}_{a,p}]$ is equal to 1 iff actor a acts on the overlapping area $\mathcal{A}^m_{c,ov}$ using power level $P^{(m)}_{a,p}$.
- $T^{(m)}_{a,p}\ [s]$ is the action completion time for actor a acting alone and independently on the mth overlapping area, when the actor uses the pth power level

$$T^{(m)}_{a,p} = K \cdot \frac{A^m_{c,ov}}{\eta_a \cdot (P^{(m)}_{a,p})^{\gamma_a}}, \tag{13}$$

  where $K\ [W^{\gamma_a} \cdot s/m^2]$ is a constant, γ_a is a parameter ranging in (0, 1], which defines the power-time relationship for actor a, and η_a is the actor efficiency.
- $\delta\ [s]$ is the *action completion bound* (that is, the maximum time for the action to be completed), which depends on the event and on the application.
- $I^{(m)}_a$ is equal to 1 iff the mth overlapping area is in the action range of actor a; otherwise, it is equal to 0.
- h_a is a binary variable equal to 1 iff actor a is involved in an action.

We can now formulate the optimization problem as follows:

$\mathbf{P}^{Res}_{Max}$: **Residual Energy Maximization Problem**

$$Given: \quad N_A, L, M_c, E^{Av}_a, T^{(m)}_{a,p}, I^{(m)}_a$$

$$Find: \quad \underline{X}^{(m)} = [x^{(m)}_{a,p}], h_a \tag{14}$$

$$Maximize: \quad E^{Res}_{Avg} = \frac{\sum_{a=1}^{N_A} h_a E^{Res}_a}{\sum_{a=1}^{N_A} h_a} \tag{15}$$

$$Subject\ to:$$

$$E^{Res}_a = E^{Av}_a - E^{Req}_a \geq 0, \forall a, \tag{16}$$

$$E^{Req}_a = \sum_{m=1}^{M_c} \left(\frac{\sum_{p=1}^{L} x^{(m)}_{a,p} P^{(m)}_{a,p}}{\sum_{a=1}^{N_A} \sum_{p=1}^{L} \frac{x^{(m)}_{a,p}}{T^{(m)}_{a,p}}} \right), \forall a, \tag{17}$$

$$\sum_{p=1}^{L} x^{(m)}_{a,p} \leq 1, \forall a, \forall m, \qquad \sum_{a=1}^{N_A} \sum_{p=1}^{L} x^{(m)}_{a,p} \geq 1, \forall m, \tag{18}$$

$$\frac{1}{\sum_{a=1}^{N_A} \sum_{p=1}^{L} \frac{x^{(m)}_{a,p}}{T^{(m)}_{a,p}}} \leq \delta, \forall m, \tag{19}$$

$$h_a \leq \sum_{p=1}^{L} \sum_{m=1}^{M_c} x^{(m)}_{a,p}, \forall a, \qquad h_a \geq x^{(m)}_{a,p}, \forall a, \forall p, \forall m, \tag{20}$$

$$x^{(m)}_{a,p} \leq I^{(m)}_a, \forall a, \forall p, \forall m. \tag{21}$$

Constraint (16) guarantees a nonnegative residual energy for each actor. Constraint (17) defines the energy required

for actor a to complete the action on the overlapping areas where it is involved. The constraints in (18) ensure that each actor uses only one among its power levels and that at least one actor acts on each overlapping area, respectively. Note that, when multiple actors act on an area, the time to complete the action is reduced. Accordingly, constraint (19) limits the overall action completion time expressed as

$$\left(\sum_{a=1}^{N}\sum_{p=1}^{L}\frac{x_{a,p}^{(m)}}{T_{a,p}^{(m)}}\right)^{-1}$$

for multiple actors acting on the area to be smaller than the action completion bound for each overlapping area. The constraints in (20) define the relation between the $x_{a,p}^{(m)}$ and h_a variables, whereas constraint (21) imposes that each actor act only on areas in its action range.

6 Actor-Actor Coordination: Localized Auction Protocol

In this section, we propose a distributed solution to the actor-actor coordination problem stated in Section 5. Our solution is inspired by the behavior of agents in a *real-time auction* [21], [22] and describes the behavior of actors as agents participating in transactions as buyers/sellers. The objective of the auction is to select the best set of actors to perform the action on each overlapping area. Thus, overlapping areas can be seen as *items* that are traded by the actors. Actors can assume the following roles:

- *Seller.* The actor responsible for a portion of an event area, that is, the actor that receives event features for that area. It corresponds to a collector.
- *Auctioneer.* The actor in charge of conducting the auction on a particular overlapping area. It is selected for each overlapping area by the collector/seller responsible for that area.
- *Buyer.* The actors that can act on a particular overlapping area.

A localized auction takes place in each overlapping area. The *bid* of each actor participating in the auction consists of a power level and the corresponding *action completion time* (that is, the time needed by that actor to complete the action on the whole area) defined in (12) and (13), respectively, as well as the available energy of the actor. The objective is to maximize the total *revenue* of the team, where the team is constituted by the actors participating in the auction and the revenue depends on the *residual energy* (that is, E_{Avg}^{Res} as given in Section 5). Multiple localized auctions take place in parallel under the responsibility of different auctioneers. This is preferable to one single auction conducted by the collector for several reasons: 1) it causes lower signaling overhead since the auction messages are exchanged between the auctioneer and the buyers for that overlapping area, which are close to the auctioneer, 2) the auction process workload is shared among a higher number of actors since the number of auctioneers is, in general, higher than the number of collectors, and 3) it is scalable, as the number of actors increases.

When seller c (the collector) receives the event features from the sensors, it decides whether an action needs to be performed on the area that it is responsible for and computes all the nonoverlapping and overlapping areas. The coordination problem arises for the overlapping areas where more than one actor can act, whereas, for the nonoverlapping areas, the seller directly assigns the action task to the corresponding actor.

Seller c selects M_c *auctioneers*, one for each overlapping area, among the actors that can act on each of these areas. Let $s^{(m)} \in \mathcal{S}^A$ be the auctioneer selected by seller c to conduct the auction for the mth overlapping area. This auctioneer is selected to be the closest actor to the center of the overlapping area. In this way, since the auctioneer is close to each actor in the overlapping area, the energy spent for communication and the auction time are reduced. After selecting the auctioneer $s^{(m)}$, the seller c provides it with the area $\mathcal{A}_{c,ov}^{m}$ where the auction should take place, the *action completion bound* δ, and the *auction time bound* τ_c, which is the maximum allowed time for the auction. The auctioneer determines the winners of the auction based on the bids that it receives from the buyers. At the beginning of the auction, the auctioneer sends a `JOIN_AUCTION` message to all the buyers competing for the area. After a buyer a hears this announcement, it submits its available energy, E_a^{Av}, and L 2D bids $\underline{b_a} = \{b_a^1, b_a^2, \ldots, b_a^L\}$, where $b_a^{(p)} = [P_{a,p}^{(m)}; T_{a,p}^{(m)}]$ and $p = 1, 2, \ldots, L$, with $P_{a,p}^{(m)}$ and $T_{a,p}^{(m)}$ defined in (12) and (13), respectively. By means of these bids, the auctioneer determines the winners by calculating the optimal solution for the residual energy maximization problem $\mathbf{P}_{Max}^{Res}$ defined in Section 5. However, in this case, the problem is limited to the overlapping area the auctioneer is responsible for. In this way, since the bids are submitted to the auctioneer only once, the signaling overhead is reduced [23]. In microeconomic theory, this auction mechanism can be classified as a *single-round sealed-bid auction* [21], where each buyer submits its bids in one shot irrespective of the bids from other buyers.

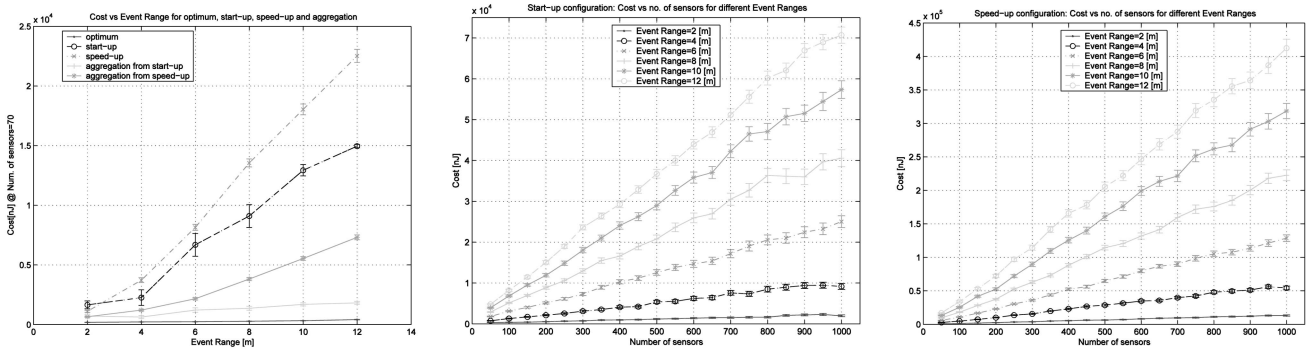
The solution proposed in this section can also be seen as a "divide-and-conquer" approach to the problem discussed in Section 5. Each "auctioneer" actor solves a smaller scale version of the original problem. In this way, several subproblems are solved in parallel and independently. However, as will be shown in Section 7, the attained performance is much better than that of simpler heuristics and, in general, very close to the global optimum.

7 Performance Evaluation

In this section, we evaluate the performance of the proposed framework. In Sections 7.1 and 7.2, we report the performance results for the sensor-actor coordination, whereas, in Section 7.3, we discuss the actor-actor coordination.

7.1 Sensor-Actor Coordination

The optimization problem presented in Section 3.2 was implemented in A Mathematical Programming Language (AMPL) [24] and solved with CPLEX [25]. The start-up, speed-up, and aggregation states described in Section 4 were implemented in a C++ simulator, which we used to evaluate the energy consumption and, in the J-Sim simulator [26], which implements the whole protocol stack of a sensor node. The figures in this section report 95 percent confidence intervals. We considered several different simulation scenarios. In Scenario 1, the deployment area is circular, with a radius equal to 20 m. For



(a) (b) (c)

Fig. 4. (a) Scenario 1. Comparison of optimal solution, speed-up, start-up, and aggregation configurations with 70 nodes. (b) Start-up configuration: energy consumption versus the number of sensors for different event ranges. (c) Speedup configuration: energy consumption versus the number of sensors for different event ranges.

each deployed sensor, the distance from the center of the area and the angle are uniformly distributed random variables. In Scenario 2, sensor nodes are randomly deployed in a square area of 25 $\times$ 25 m. The event area is circular, with varying radi ranging in [2, 12] m in different simulations. The epicenter of the event area is randomly selected such that the event area completely falls into the terrain. Scenario 3 is similar to Scenario 2, but the side of the square area is 100 m. Four actors are randomly deployed in each scenario. As in [10], the simulation parameters for the energy model in Section 3.1 are chosen to be $E_{elec} = 50\ \mathrm{nJ/bit}$, $\beta = 100\ \mathrm{pJ/bit/m}^{\alpha}$, and $\alpha = 4$. The transmission range of sensors is set to 10 m.

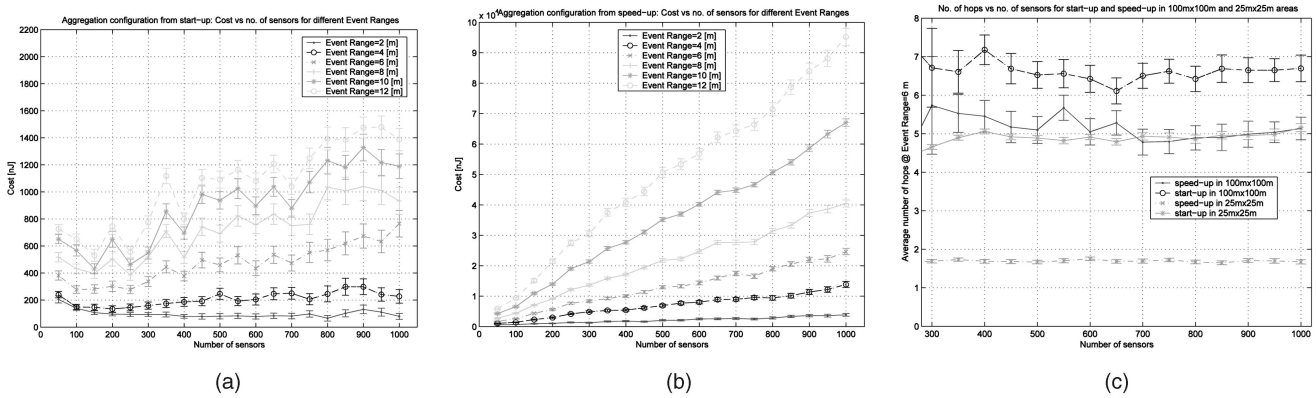
In this section, we refer to a start-up configuration, a speed-up configuration, and an aggregation configuration as the configurations where all nodes are in the start-up, speed-up, and aggregation states, respectively. This allows us to show the benefits of the proposed solution without depending on the choice of parameters that govern the transitions among states. Dynamic aspects are discussed in Section 7.2.

Fig. 4a shows a comparison between the optimal solution to the event-driven partitioning problem described in Section 3 and the energy consumption in the start-up, speed-up, and aggregation configurations in Section 4, respectively, with varying event ranges. The figure shows the overall network cost, that is, the energy needed to transmit one bit from each source to the actors. Noticeably, the optimal solution is almost independent of the event range. This is due to two contrasting phenomena. The number of sources increases when the event range increases, leading to a potentially higher energy consumption; at the same time, since more nodes are involved, aggregation can be increasingly leveraged. These two trends compensate for each other, leading to a flat curve. Conversely, the energy consumption in the start-up and speed-up configurations highly increases with the event range. As is also shown in Fig. 4a, this can be partially compensated by the aggregation state. In particular, an aggregation configuration can be reached from both a start-up configuration and a speed-up configuration. An aggregation configuration reached from a start-up configuration leads to an almost-optimal energy consumption, whereas, by reaching the aggregation configuration from a speed-up configuration, the energy consumption can still be decreased consistently, but not as much as in the previous case. Hence, Fig. 4a motivates the design of our distributed protocol. In fact, the distributed solution described in Section 4 modifies the structure of the datrees to reach an energy configuration that is between the speed-up and the aggregation from start-up curves as shown in Fig. 4a. Depending on the required latency bound and reliability threshold, after a transient start-up configuration, a certain number of sensors will enter the speed-up/aggregation state to reach a minimum energy configuration, given the required reliability.

In Figs. 4b and 5a, we plot the average energy consumption versus the number of sensors, with different event ranges, for the start-up and aggregation configurations in Scenario 2. The energy expenditure in the aggregation configuration is two orders of magnitude lower than in the start-up configuration. As can be seen in Fig. 5a, the energy expenditure increases sublinearly with the number of sensors. Fig. 4c reports the overall energy consumption for the speed-up configuration. Interestingly, not only is the energy consumption of the speed-up configuration around one order of magnitude higher than in the start-up configuration, but also, as already seen in Fig. 4a, when the aggregation configuration is reached from a speed-up configuration, the network converges to a less energy-efficient configuration, compared to when the aggregation configuration is reached directly from the start-up configuration. This is confirmed in Fig. 5b, which shows that the order of magnitude of the energy consumption is 10^4 nJ for an aggregation configuration reached from a speed-up configuration. Conversely, as shown in Fig. 5c, in Scenario 2, the average number of hops of each source-actor pair is reduced from around five hops for the start-up configuration to less than two hops in the speed-up configuration.

7.2 Convergence of DEPR

In this section, we discuss the convergence of the DEPR mechanism for sensor-actor coordination introduced in Section 4.5. The mechanism was implemented in J-Sim [26]. Each sensor in the event area is a constant bit rate (CBR) source that generates 10 packets/s. The packet size is 56 bytes. At the network layer, sensors behave according to the DEPR protocol described in Section 4. The media access control (MAC) layer is based on carrier sense multiple access with collision avoidance (CSMA/CA), whereas the



(a) (b) (c)

Fig. 5. (a) Scenario 2. Aggregation configuration reached from the start-up configuration: energy consumption versus number of sensors for different event ranges. (b) Scenario 2. Aggregation configuration reached from speed-up configuration: energy consumption versus number of sensors for different event ranges. (c) Scenarios 2-3. Average number of hops for the start-up and speed-up configurations.

physical layer in J-Sim is enhanced with the power control mechanism described in Section 4. In this simulation, 100 sensors are randomly deployed in a 100 × 100 m terrain. The maximum transmission range is set to 40 m and the capacity of the channel is set to 400 Kbps, whereas the interface queue length is set to 20 packets. The event radius is equal to 15 m and is centered in the middle of the simulation terrain. We evaluate the mechanism from the perspective of one actor that is placed in the middle of the lower side of the deployment terrain. We implemented a linear predictor that calculates the predicted reliability as $\hat{r}[k+1] = \sum_{i=0}^{R-1} a_i \cdot r[k-i]$, with $R = 2$, $a_0 = 2$, and $a_1 = -1$. A thorough analysis of the impact of different predictors on the convergence of the proposed mechanism is out of the scope of this paper. The feedback period Δf is set to 1 s and the decision interval Δd is set to 5 s, whereas the short decision interval Δd_{sh} is set to 1 s. The reliability threshold is set to $r_{th} = 0.80$, whereas the high and low reliability thresholds are set to $r_{th}^{+} = 0.90$ and $r_{th}^{-} = 0.78$, respectively. The delay bound is set to 200 ms, which is a reasonable value for several monitoring applications.

In the experiments shown, we assume an ideal feedback, that is, sensor nodes receive feedback from the actors reliably and without delay. This is to decouple the analysis of the convergence of DEPR from the particular multicast mechanism adopted. This also accurately models the situation where sensor nodes have a different radio on a different frequency to receive beacons from the actors. We have also performed experiments to assess the effect of unreliable broadcasts and of transmission delays on the mechanism, where feedback messages are transmitted by the actors and relayed by intermediate sensors until they are received by all devices in the da-trees. As expected, in this case, delays prevent the reliability from asymptotically stabilizing to the desired reliability threshold value, as in the ideal case. However, although small fluctuations occur, the average reliability lies within the high and low reliability thresholds and its minimum does not fall below r_{th}^{-}.

As previously discussed, DEPR includes several parameters that flexibly allow adapting its behavior. In general, parameter tuning is done either based on analytical models of the system or by simulation. Since analytical models that accurately model the delay of large-scale wireless sensor networks under different conditions are still largely missing or are based on restrictive assumptions, we performed it by simulation. The probabilities that govern the transitions among different states are set as follows: The probability P_{st-sp} of moving from the start-up state to the speed-up state is set to 0.5 when the advertised reliability $r < r_{th}^{-}(0.1 \cdot r_{th}^{-})$, which is a very low reliability. Otherwise, if the reliability is low but close to the threshold, then we try to smoothly increase the reliability and set $P_{st-sp} = 0.1$. The probabilities P_{st-ag} and P_{sp-ag} of moving to the aggregation state from the start-up and speed-up states, respectively, are equally set to 0.05 if the advertised reliability is equal to 1 and 0.02 otherwise. In any case, the probability of switching into the aggregation state needs to be low (less than 0.1), as higher values almost invariably cause instabilities, provoking sudden drops of the observed reliability in the transients. Finally, the transition probability P_{ag-sp} from the aggregation to the speed-up state is set to 0.2 if the current, predicted, and short-term reliabilities are all below the threshold r_{th}^{-}. Similarly, P_{ag-sp} is set to 0.1 if only the short-term and predicted reliabilities are below the threshold, whereas the current reliability is still above. Meanwhile, it is set to 0.05 if only the short-term reliability is below the threshold, whereas the others are still above. With such tuning of the parameters, our objective is to minimize the probability that the reliability drops below the threshold and still converge as quickly as possible to a lower energy configuration.

Fig. 6a shows the event reliability as observed by the actor. Immediately after the start-up state, the reliability drops below the threshold. Hence, the actor advertises low reliability and a high number of sensors move to the speed-up state. This increases the reliability above the threshold, which, in turn, causes a small portion of the sensor nodes to move to the aggregation state. After a few oscillations, the reliability stabilizes at the desired value, that is, within the high and low reliability thresholds. Fig. 6b shows the evolution of the number of sensors in each of the three active states. Fig. 6c shows the distribution of the delays during the simulation time, whereas Fig. 7a shows the evolution of the delays during the simulation time. As expected, higher delays are encountered during periods of lower reliability, and vice versa.

7.3 Actor-Actor Coordination

In this section, we discuss some performance results of the actor-actor coordination problem defined in Section 5. The MINLP problem was implemented in AMPL and solved

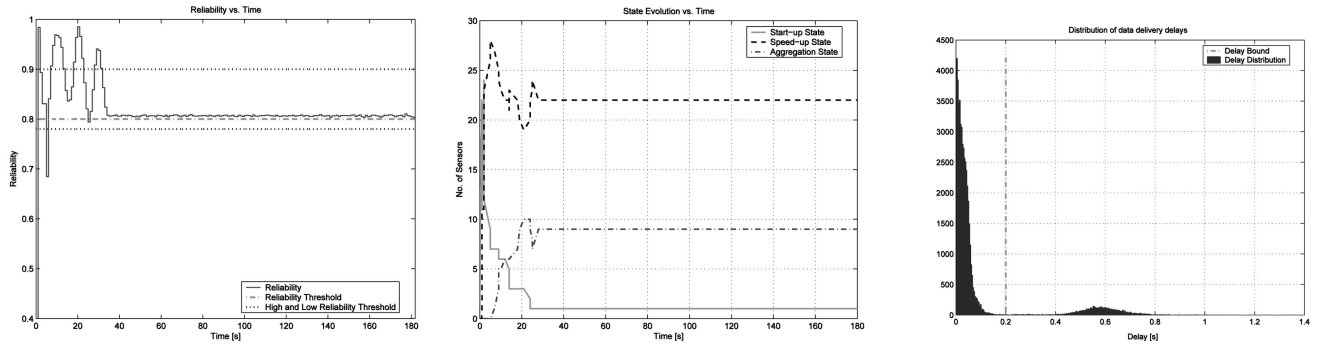


Fig. 6. (a) Convergence of DEPR: reliability of the event observed at the collector/actor. (b) Convergence of DEPR: number of sensors in each state versus time. (c) Distribution of delays.

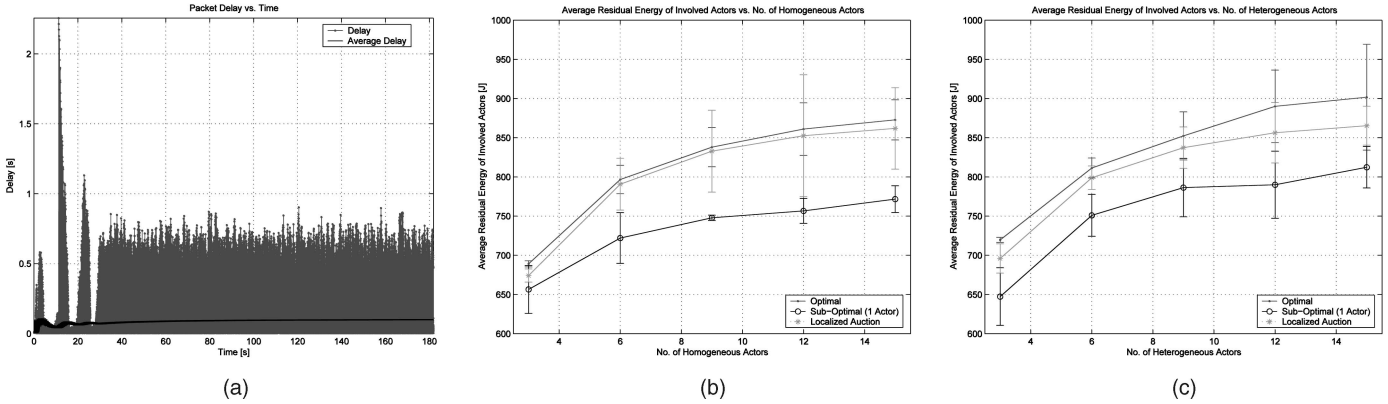


Fig. 7. (a) Delays with simulation time. (b) Average residual energy of involved actors in the homogeneous case. (c) Average residual energy of involved actors in the heterogeneous case.

with the solver available through the NEOS Optimization Server [6]. In Figs. 7b and 7c, we compare the average residual energy with three different solution approaches, namely, the *optimal*, *one-actor*, and *localized auction*. In the optimal solution, the best set of actors is chosen by solving the problem in Section 5. In the one-actor heuristic, the action is performed by one actor only for each overlapping area, that is, the actor with the highest residual energy after the completion of the action. In the localized auction, the problem is solved as explained in Section 6.

In the experiments performed, we concentrate on two scenarios with three overlapping areas: one with homogeneous actors, with $\gamma_a = 0.8$ (Fig. 7b), and one with heterogeneous actors, half of which have $\gamma_a = 0.6$ (low-efficiency actors) and the other half have $\gamma_a = 0.9$ (high-efficiency actors; Fig. 7c). For the remaining parameters defined in Section 5, we assume the following values: $A^1_{c,ov} = 50\ \mathrm{m}^2$, $A^2_{c,ov} = 100\ \mathrm{m}^2$, $A^3_{c,ov} = 150\ \mathrm{m}^2$, $P_a^{Max} = 100\ \mathrm{W}$, $L = 5$, $K/\eta_a = 1\ \mathrm{W}^{\gamma_a} \cdot \mathrm{s/m}^2$, and $\delta = 10\ \mathrm{s}$. The value of the initial available energy E_a^{Av} of an actor is a random variable uniformly distributed between 800 and 1,000 J.

As shown in both Figs. 7b and 7c, the localized auction mechanism leads to near-optimal residual energy, as each auctioneer calculates the optimal solution separately for its overlapping area. However, this greatly simplifies the problem and can be achieved with local communications among actors. Moreover, in the heterogeneous scenario, the proposed localized solution effectively exploits the high-efficiency actors, thus reducing the dissipated energy to complete the action.

8 Conclusions

We presented a coordination framework for WSANs and discussed the sensor-actor and actor-actor coordination problems. We developed an optimal solution for the sensor-actor coordination based on an event-driven partitioning paradigm and formulated it as an ILP. We also proposed DEPR, a distributed protocol for sensor-actor coordination that includes an adaptive mechanism to trade off energy consumption for delay when the event data has to be delivered to the actors within predetermined latency bounds. For the actor-actor coordination, an optimization model was defined for a class of coordination problems in which the area to be acted upon is optimally split among different actors. The problem was formulated as an MILP and an auction-based localized solution of the problem was also presented.

Acknowledgments

A preliminary shorter version of this paper was presented at ACM MobiHoc 2005, Urbana-Champaign, Illinois. This material is based upon work supported by the US 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,* vol. 2, no. 4, pp. 351-367, Oct. 2004.

[2] 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 in Comm.,* vol. 23, no. 3, pp. 520-532, Mar. 2005.

[3] W.-P. Chen, J.C. Hou, and L. Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 3, no. 3, pp. 258-271, July-Sept. 2004.

[4] X. Ji, H. Zha, J. Metzner, and G. Kesidis, "Dynamic Cluster Structure for Object Detection and Tracking in Wireless Ad Hoc Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '04),* June 2004.

[5] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Feb. 1993.

[6] J. Czyzyk, M. Mesnier, and J. Moré, "The NEOS Server," *IEEE J. Computational Science and Eng.,* vol. 5, no. 3, pp. 68-75, July-Sept. 1998.

[7] R. Vedantham, Z. Zhuang, and R. Sivakumar, "Hazard Avoidance in Wireless Sensor and Actor Networks," *Computer Comm.,* vol. 29, nos. 13-14, pp. 2447-2736, Aug. 2006

[8] 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 '03),* pp. 46-55, May 2003.

[9] 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. INFOCOM,* Mar. 2005.

[10] 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.

[11] O. Younis and S. Fahmy, "Distributed Clustering in Ad Hoc Sensor Networks: A Hybrid Energy-Efficient Approach," *Proc. INFOCOM,* Mar. 2004.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing Newly Deployed Ad Hoc and Sensor Networks," *Proc. MobiCom,* Sept. 2004.

[13] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, no. 4, pp. 393-422, Mar. 2002.

[14] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, pp. 51-58, May 2000.

[15] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co., 1979.

[16] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock Synchronization for Wireless Sensor Networks: A Survey," *Ad Hoc Networks,* vol. 3, no. 3, pp. 281-323, May 2005.

[17] G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Technical Report ISI ISU/RR-87-180, Mar. 1987.

[18] R.W.F. Kuhn and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad Hoc Routing," *Proc. MobiHoc,* June 2003.

[19] 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.

[20] J. Na and C. Kim, "GLR: A Novel Geographic Routing Scheme for Large Wireless Ad Hoc Networks," *Computer Networks,* vol. 50, no. 17, pp. 3225-3522, Dec. 2006.

[21] R.P. McAfee and J. McMillan, "Auctions and Bidding," *J. Economic Literature,* vol. 25, no. 2, pp. 699-738, June 1987.

[22] B.P. Gerkey and M.J. Mataric, "Sold!: Auction Methods for Multirobot Coordination," *IEEE Trans. Robotics and Automation,* vol. 18, no. 5, pp. 758-768, Oct. 2002.

[23] P. Maillé and B. Tuffin, "Multi-Bid Auctions for Bandwidth Allocation in Communication Networks," *Proc. INFOCOM,* Mar. 2004.

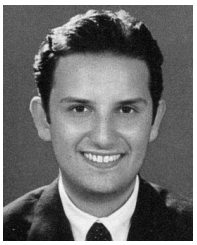
[24] R. Fourer, D.M. Gay, and B.W. Kernighan, *AMPL: A Modeling Language for Math. Programming.* Duxbury Press/Brooks/Cole Publishing, 2002.

[25] CPLEX Solver, http://www.cplex.com, 2006.

[26] The J-Sim Simulator, http://www.j-sim.org/, 2006.

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. He received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in 2007 after working at the Broadband and Wireless Networking Laboratory (BWN-Lab) advised by Prof. I.F. Akyildiz. In 2007, he joined the Electrical Engineering Department at the University at Buffalo, State University of New York, as an assistant professor. His research interests are in the design, modeling, and optimization of ad hoc and sensor networks with an emphasis on wireless sensor and actor networks, multimedia sensor networks, and underwater acoustic sensor networks. He is the recipient of the 2004 BWN-Lab Researcher of the Year Award. He is a student 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. He received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in June 2007 after working at the Broadband and Wireless Networking Laboratory (BWN-Lab) under the direction of Prof. I.F. Akyildiz. In 2005, he was awarded the BWN-Lab Researcher of the Year award for "outstanding contributions and professional achievements." In September 2007, he joined the Electrical and Computer Engineering Department at Rutgers, The State University of New Jersey, as an assistant professor. His research interests are in ad hoc and sensor networks, underwater acoustic communications, wireless sensor and actor networks, and network optimization and control. He is a student member of the IEEE.



Vehbi C. Gungor received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in June 2007 after working at the Broadband and Wireless Networking Laboratory (BWN-Lab) under the direction of Prof. I.F. Akyildiz. Currently, he is a communication system architect at Eaton Corporation. His current research interests are in wireless ad hoc and sensor networks, wireless mesh networks, WiMAX, IP networks, and electric utility automation. He is a student member of the IEEE.

Ian F. Akyildiz is the Ken Byers Distinguished Chair Professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology, and the director of Broadband and Wireless Networking Laboratory. He is the editor in chief of *Computer Networks* (Elsevier) and the *Ad Hoc Networks Journal* (Elsevier). He served as a national lecturer for ACM from 1989 until 1998. His current research interests include next-generation wireless networks, sensor networks, and wireless mesh networks. He is a fellow of the IEEE (1995) and the ACM (1996). He received the ACM Outstanding Distinguished Lecturer Award in 1994, the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society) for his paper titled "Multimedia Group Synchronization Protocols for Integrated Services Architectures" published in the *IEEE Journal of Selected Areas in Communications* (JSAC) in January 1996, and the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He also received the 2003 IEEE Best Tutorial Award (IEEE Communications Society) for his paper titled "A Survey on Sensor Networks" published in *IEEE Communication Magazine* in August 2002 and the 2003 ACM SIGMOBILE Award for his significant contributions to mobile computing and wireless networking.