

A distributed activity scheduling algorithm for wireless sensor networks with partial coverage

Tarik Yardibi · Ezhan Karasan

Published online: 1 August 2008
© Springer Science+Business Media, LLC 2008

Abstract One of the most important design objectives in wireless sensor networks (WSN) is minimizing the energy consumption since these networks are expected to operate in harsh conditions where the recharging of batteries is impractical, if not impossible. The sleep scheduling mechanism allows sensors to sleep intermittently in order to reduce energy consumption and extend network lifetime. In applications where 100% coverage of the network field is not crucial, allowing the coverage to drop below full coverage while keeping above a predetermined threshold, i.e., partial coverage, can further increase the network lifetime. In this paper, we develop the distributed adaptive sleep scheduling algorithm (DASSA) for WSNs with partial coverage. DASSA does not require location information of sensors while maintaining connectivity and satisfying a user defined coverage target. In DASSA, nodes use the residual energy levels and feedback from the sink for scheduling the activity of their neighbors. This feedback mechanism reduces the randomness in scheduling that would otherwise occur due to the absence of location information. The performance of DASSA is compared with an integer linear programming (ILP) based centralized sleep scheduling algorithm (CSSA), which is devised to find the maximum number of rounds the network can survive assuming that the location information of all sensors is available. DASSA is also compared with the decentralized DGT algorithm. DASSA attains network lifetimes up to 92% of the centralized solution and it

achieves significantly longer lifetimes compared with the DGT algorithm.

Keywords Wireless sensor networks · Energy efficiency · Sleep/activity scheduling · Partial coverage

1 Introduction

Wireless sensor networks (WSN) consist of sensor devices deployed across a geographic area to sense the environment by measuring physical parameters such as temperature, motion, etc. [1]. The most important issue regarding the design of WSNs is the energy consumption since these networks are usually deployed in remote areas where the replacement of batteries is impossible.

The lifetime of a WSN can be significantly increased by allowing some of the nodes to sleep intermittently. *Activity scheduling* or *sleep scheduling* controls the number of sensors which are sensing, receiving and transmitting data, such that some user defined constraints are satisfied. While ensuring only a subset of nodes to be in the operating mode, the sleep scheduling mechanism must fulfill two requirements: *connectivity* and *coverage*. A WSN is connected if each active sensor in the network can reach the sink, possibly via multiple hops. Coverage is defined as the area that can be monitored by the active sensors that can reach the sink.

A detailed survey of the sleep scheduling algorithms has been recently conducted in [2]. Sleep scheduling algorithms can be categorized into four groups. In the first group, active nodes only provide connectivity to the network without considering coverage. GAF [3], SPAN [4], ASCENT [5], S-MAC [6] and PEAS [7] protocols are examples to this group. In the second group, the nodes are

T. Yardibi · E. Karasan (✉)
Electrical and Electronics Engineering Department,
Bilkent University, 06800 Ankara, Turkey
e-mail: ezhan@ee.bilkent.edu.tr

T. Yardibi
e-mail: ytarik@ee.bilkent.edu.tr

selected to be active so as to assure full coverage of the sensor field without considering connectivity. A linear programming approach is used in [8] to determine the minimum number of sensors which can fully cover a certain area. In [9, 10] nodes whose sensing areas are covered by their neighbors are turned off with a random backoff mechanism. [11] extends this work and reduces the redundancy among the active nodes.

Finding the minimum set of connected sensors which cover the entire deployment area is proven to be an NP-hard problem [12]. However, there are many heuristic algorithms for providing coverage and connectivity. Some algorithms select mutually exclusive sets of sensor nodes, where the members of each of those sets together completely cover the monitored field. These sets are used in a round robin fashion in order to increase the lifetime while maintaining the coverage [13, 14]. In CCP [15] and OGDC [16], it is proven that if the radio range is at least twice the sensing range, complete coverage of a network field guarantees connectivity. CCP schedules nodes to sleep depending on the coverage degrees of the intersection points of a node's sensing disk with its neighbors' sensing disks. In OGDC, the optimum location of a third disk to cover the intersection points of two other disks, with minimum overlapping sensing areas, is found. Then, OGDC tries to schedule nodes which are close to these optimum locations to be active. Both CCP and OGDC require location knowledge of sensors.

In some applications, covering 100% of the sensor field continuously is not necessary. Instead, the network lifetime can be prolonged if the coverage level is kept below 100%. For example, in a temperature or humidity monitoring WSN, it may be sufficient to cover just 90% of the total sensor field in order to increase network lifetime. This is named as partial coverage. This concept is fairly new and is not extensively studied in the literature. The last group of existing work on activity scheduling exploits partial coverage. The theoretical bounds for the number of active nodes to satisfy a certain coverage level while being connected is derived in [17]. pCover [18] is a distributed algorithm for the partial coverage problem assuming all the nodes are aware of their locations and their neighbors' locations. Nodes are scheduled for sleep according to the coverage loss incurred by their deactivation in [19], where the locations of the sensors are assumed to be known. An upper bound for the network lifetime when only a portion of the sensor field is to be covered, assuming that the deployed nodes form a homogenous Poisson point process, is found in [20].

In [21], a distributed scheduling mechanism is proposed which does not assume location information while providing statistical guarantees on the sensing coverage. However, this algorithm does not guarantee connectivity. The distributed scheduling algorithm in [22], which we

name data gathering tree (DGT) algorithm hereafter, also does not assume location information. DGT exploits partial coverage and involves similar design specifications used in this paper. The minimum number of sensors, k , to provide a certain coverage level is found by using geometric probability theory assuming that the nodes are distributed uniformly. Network operation is divided into reporting cycles such that the period for each sensor to report its data is equal to the duration of these reporting cycles. Each cycle is further divided into δ slots, where $\delta = \lfloor \frac{N}{k} \rfloor$ and N is the number of sensors. Each sensor randomly assigns itself a slot in $\{1, \dots, \delta\}$ and reports its data only at that slot in each reporting cycle. This way, disjoint sets of sensors are found that provide the required coverage on the average. In order to ensure connectivity, it may become necessary to use non-disjoint sets. To find k , the distribution of the sensors on the field has to be known and the integrations over these distributions have to be tractable. A drawback of this algorithm is that the scheduling messages are large in size which increases the energy consumption.

Although most of the work in the literature consider network lifetime, connectivity and coverage, there are other studies on sleep scheduling that also consider the delay in data gathering for delay sensitive sensor networking applications. Pipelined tone wakeup (PTW) is proposed in [23] which achieves a balance between energy saving and end-to-end delay. PTW is based on an *asynchronous* wakeup scheduling mechanism that overlaps the wakeup procedures with the packet transmissions. During a transmission, the receiver node wakes up all its neighbors using the signaling channel, which makes it possible to forward the data to the next hop as soon as the current transmission is completed, reducing the end-to-end delay. To accomplish this, PTW uses a second wireless interface, a signaling radio, which is used to awaken a node when it is needed for data transmission. The assumption is that the power consumption of this secondary radio is extremely low, and thus it can remain active at all times without having a major impact on the lifetime of the device. Although there are low power radio technologies today, their power consumptions are not negligible and their transmission ranges are much shorter than the ranges of technologies used for the primary wireless interface.

In this work, we assume that there is a single radio interface at each sensor node, and a *synchronous* activity scheduling mechanism is used where nodes periodically wake up at the same time in order to determine their schedules for the next scheduling interval. It is also assumed that the sensor data is *periodically* generated by the sensor nodes and gathered at the sink node. The main objective of the proposed distributed adaptive sleep scheduling algorithm (DASSA) is to maximize the network lifetime subject to connectivity and partial coverage

constraints, without requiring the location information of the sensor nodes. Other metrics, such as the latency in data gathering, are not explicitly considered by DASSA. However, the delay in DASSA can be adjusted by properly choosing the duration of the scheduling interval.

The main contributions of this paper are as follows. An integer linear programming (ILP) based centralized sleep scheduling algorithm (CSSA) which uses location information is formulated to provide an upper bound for the lifetime achieved by the distributed sleep scheduling algorithms. Next, a distributed scheduling algorithm, DASSA, which requires no location information and exploits partial coverage is proposed. The set of nodes selected to be active by DASSA are always connected due to the nature of the algorithm. Simulation results show that DASSA performs close to the performance of the centralized algorithm. Comparisons with DGT show that the network lifetime can be significantly extended by DASSA especially for relatively low partial coverage targets.

In the rest of the paper, we first describe the network model in Sect. 2. The centralized scheduling algorithm is explained in Sect. 3. We then present the design and analysis of DASSA in Sect. 4. Extensive simulation results are presented in Sect. 5 and the paper is concluded in Sect. 6.

2 Network model

Wireless sensor networks is modeled by an undirected connected graph $G = (V, E)$ where $V = \{V_0, V_1, \dots, V_N\}$ represents all the sensor nodes in the network including the sink V_0 and E represents the communication links between these nodes. For any node i and j , the edge $(i, j) \in E$ if and only if i and j are within the communication range of each other. A sensor node can communicate with all its neighbors in its transmission range, R_t , and can sense every event occurring within its sensing range, defined as the disk with radius R_s . We assume that each sensor node has the same initial energy, E_{init} . These assumptions are widely used in many sensor networking algorithms in the literature including [15, 16, 22].

The data reporting model is assumed to be time-driven [16, 22]; the network operation is divided into rounds and all active sensor nodes are responsible for transmitting their data together with their descendants' data towards the sink at each round. The sleep scheduling algorithm is executed at the beginning of each round and nodes forward data accordingly. This reporting model requires time synchronization among the nodes. One choice for this purpose can be the S-MAC [16, 22] protocol which provides both time synchronization and additional energy savings in the MAC layer. Many other alternatives are available for time synchronization, e.g., [24, 25] and as a MAC protocol, e.g., [26].

We assume that the WSN is uniformly deployed on a square field and the sink is located at the center of the field. The field is divided into G square grids. The sink is assumed to have abundant energy resources and is responsible for gathering the data of all the active sensors and then transmitting it to a centralized location, e.g., through satellite.

If node i is in the transmission range of node j , then node i and j are at a single hop distance to each other. The tier number of a node is the minimum number of hops required to reach the sink from that node, i.e., a node which can reach the sink in at least k hops belongs to tier k .

We make the following definitions:

- Grade of coverage (GoC) is the desired partial coverage level such that $0 \leq \text{GoC} \leq 1$. GoC is calculated as the ratio of the number of grids covered by the set of active sensors that can communicate with the sink to the total number of grids.
- Grade of coverage lifetime (GoC-L) is the number of rounds for which the coverage provided by the network satisfies GoC.

3 Centralized sleep scheduling

Centralized sleep scheduling algorithm is an ILP based algorithm which assumes global knowledge of sensor locations and energies. This algorithm is not suitable to be used in a distributed scenario and the main purpose of devising such an algorithm is to calculate the lifetime that can be achieved by a centralized approach which uses the global information of the WSN. CSSA will be used in Sect. 5 for evaluating the performances of the distributed scheduling algorithms.

As intuition suggests, the nodes which are close to the sink are the first ones to die in the network since they have to forward both their own and their descendants' data [27, 28]. Consequently, CSSA tries to give the most emphasis to tier 1 nodes and then to tier 2 nodes and so on. For this purpose, we define a sorted ID vector, $\mathbf{SID} = \{SID_1, SID_2, \dots, SID_N\}$, which consists of the node IDs of the nodes sorted first with respect to their tier numbers and then with respect to their residual energy levels. An example \mathbf{SID} vector is given as

$$\mathbf{SID} = \{ \underbrace{1, 6}_{\text{Tier1}}, \underbrace{4, 3, 8}_{\text{Tier2}}, \underbrace{2, 5}_{\text{Tier3}} \},$$

where $e_1 < e_6$, $e_4 < e_3 < e_8$, $e_2 < e_5$ and e_i is the residual energy of node i . This algorithm exploits the empirical observations that the nodes closer to the sink are more critical for network lifetime and the residual energies of sensor nodes should be used in a balanced manner.

Centralized sleep scheduling algorithm performs lexicographic optimization where the residual energy of the

first node in **SID**, SID_1 , is maximized first. In other words, the active set of nodes is selected such that with this selected configuration, the residual energy of the first node in **SID** after the round ends is decreased as small as possible. Then, the result of this maximization is added as an additional constraint to the ILP formulation, i.e., the set of solutions is limited to those with the residual energy of the first node does not fall below the maximum value. Using this modified ILP, the energy of the second node in **SID**, SID_2 , is maximized and a similar constraint is added to the ILP formulation for limiting the residual energy of the second node. This process continues by sequentially maximizing the residual energies of all the nodes in the network.

Once this procedure is completed, the algorithm moves to the next round after the residual energies of all nodes are updated according to the scheduling result of the previous round. The algorithm is implemented until no feasible solution can be found, i.e., no subset of nodes with sufficient residual energy for satisfying the connectivity and coverage constraints can be found at some round. The GoC-L is given by the number of rounds until infeasibility.

The ILP formulation used at each step of CSSA is described next. Let T_i denote the tier number of sensor i , $1 \leq i \leq N$. In CSSA, a node can only forward data to a node with tier number less than its tier number, i.e.,

$$f_{ij}^n = 0 \quad \text{if } T_i \leq T_j, \quad \forall i, j, n, \quad (1)$$

where $f_{ij}^n = 1$ if there is a data packet transmitted from sensor i to sensor j for the data generated at sensor n and $f_{ij}^n = 0$, otherwise.

The following constraint ensures the conservation of flow. The decision variable s_i is defined as $s_i = 0$ if sensor $i, 1 \leq i \leq N$, is sleeping, and $s_i = 1$ if it is active.

$$\sum_j f_{ij}^n - \sum_j f_{ji}^n = \begin{cases} s_n, & i = n \\ -s_n, & i = \text{sink}, \forall n, i. \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Since no flow can be relayed through a sleeping node,

$$f_{ij}^n \leq s_i, f_{ji}^n \leq s_i \quad \forall n, i, j. \quad (3)$$

Let E_t and E_r correspond to the amount of energy consumed for each transmission and reception, respectively. An additional energy, E_s , is consumed for the exchange of messages used for routing and scheduling. The relationship between e_i and e'_i , that correspond to the energies of sensor i at the beginning and end of the current round, respectively, is given by

$$e'_i = e_i - \sum_n \sum_j f_{ij}^n \cdot E_t - \sum_n \sum_j f_{ji}^n \cdot E_r - E_s, \quad \forall i. \quad (4)$$

The energy of each node at the end of a round cannot become negative, i.e.,

$$e'_i \geq 0, \quad \forall i. \quad (5)$$

For calculating the coverage, we divide the network field into G grids. The indicator function $c_{ni} = 1$ if and only if grid i is within the sensing range of sensor n . The decision variable v_i is defined as $v_i = 1$ if and only if grid $i, 1 \leq i \leq N$, is covered by an active sensor. If all the nodes covering grid i are asleep, then grid i is not covered, i.e.,

$$v_i \leq \sum_{n: c_{ni}=1} s_n, \quad \forall i. \quad (6)$$

If any of the nodes whose sensing area contains grid i is active, then grid i is covered, i.e.,

$$v_i \geq c_{ni}s_n, \quad \forall i, n. \quad (7)$$

The final constraint ensures that the target partial coverage level is achieved:

$$\sum_i v_i \geq \text{GoC} \times G. \quad (8)$$

The objective function of this ILP formulation is to maximize the residual energy of i -th node in the sorted list, SID_i , at the end of the round, i.e.,

$$\text{Maximize } e'_{SID_i}. \quad (9)$$

The pseudocode for CSSA is given in Algorithm 1. *ILP-CSSA* returns the optimum solution of the ILP having the objective function (9) and the constraints (1–8) in addition to the constraints added as the iterations proceed.

4 Distributed adaptive sleep scheduling algorithm (DASSA)

Distributed adaptive sleep scheduling algorithm schedules sensors to sleep while ensuring the connectivity of the operating nodes and keeping the coverage above GoC without any knowledge of the sensor locations. DASSA is simple to implement and it extends network lifetime by exploiting partial coverage. DASSA focuses on the nodes closer to the sink, since all the network traffic has to pass through these nodes, by carefully scheduling the activity of these nodes depending on their residual energies and number of neighbors. For this purpose, the sink solves a simple ILP problem to determine the schedules of the tier 1 nodes. Since the sink typically has plentiful amount of computation resources, using it to intelligently schedule tier 1 nodes is reasonable.

A. Description of DASSA: Distributed adaptive sleep scheduling algorithm consists of four sequential steps.

Step 1: Neighbor discovery: In the first step, the sink transmits a discovery message which contains a tier number 0. A node which receives a message containing tier number i sets its own tier number to $i + 1$ and transmits its

Algorithm 1 CSSA

```

round ← 0
stop ← FALSE
V' ← V
While (!stop) do
    Calculate the tier numbers for all nodes in V'
    Form SID composed of all nodes in V'
    for i = 1 to |V'| do
        res_energy ← ILP-CSSA
        If ILP-CSSA is feasible then
            Add the following constraint to ILP-CSSA:
                e'_{SID_i} ≥ res_energy
        else
            stop ← TRUE
        end
    end
    round++
    Update residual node energies {e'_i} for all nodes i ∈ V'
    V' ← V' - {i: e'_i < E_r + E_s}
    Remove all additional constraints from ILP-CSSA except (1)–(8)
end
GOC-L ← round
    
```

node ID, tier number and residual energy level after a sufficiently long timeout in order to make sure that it receives broadcast messages from all its neighbors belonging to tier *i*. This way, all the nodes discover their tier numbers and local neighborhoods while the sink learns the IDs of the tier 1 nodes.

Step 2: Scheduling tier 1 nodes: After the neighbor discovery step ends, nodes in tier 1 broadcast the IDs of their neighbors belonging to tier 2. This broadcasting is only done in the first round. Following the common assumption of immobility for the sensors [2], this process is not repeated at each round.

Once the sink has knowledge of tier 1 nodes and their neighbors from tier 2, it schedules tier 1 nodes by using a simple ILP problem in order to find the minimum number of active nodes from tier 1 with maximum remaining energy such that these nodes are connected to α portion of the nodes from tier 2. This step is a quite important step in which tier 1 nodes are scheduled in a balanced way without using any location information as illustrated in Fig. 1.

The corresponding objective function is given by

$$\text{Minimize } \sum_i s_i - \epsilon \sum_i s_i \cdot e_i, \tag{10}$$

where $s_i = 1$ if and only if sensor *i* from tier 1 is active and e_i is the residual energy of sensor *i*. The second term in the objective function makes sure that when multiple solutions achieving the minimum number of active tier 1 nodes exist,

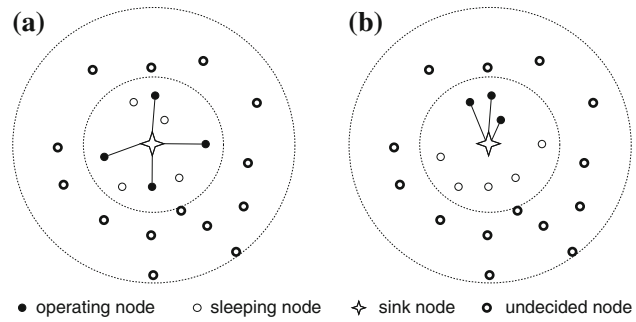


Fig. 1 The balanced scheduling of tier 1 nodes. The sink schedules tier 1 nodes which are shown in full circles. (a) Selection of DASSA. (b) Random selection

the sink chooses the solution with the maximum total residual energy. ϵ is chosen such that the objective function does not become negative but so that the residual energies still affect the solution. This value can be increased or decreased depending on the application. We empirically found out $\epsilon = 0.1/E_{init}$ to give satisfactory results in all of our simulation results.

The following two constraints are for the connectivity of the tier 2 nodes and the active tier 1 nodes. The indicator $c_{ij} = 1$ if and only if sensor *j* from tier 2 is in the communication range of sensor *i* from tier 1. $l_j = 1$ if and only if node *j* from tier 2 is active and it is connected to some active node from tier 1:

$$s_i c_{ij} \leq l_j, \quad \forall i, \tag{11}$$

$$l_j \leq \sum_i s_i c_{ij}, \quad \forall j. \tag{12}$$

The last constraint determines what percentage of nodes from tier 2 is to be connected with the active nodes in tier 1:

$$\sum_i l_i \geq \alpha n, \tag{13}$$

where n is the number of nodes in tier 2 and $0 \leq \alpha \leq 1$.

After solving the ILP problem given by (9–12), the sink broadcasts a packet containing the node IDs of the tier 1 nodes scheduled to be active. A tier 1 node receiving this message enters sleep mode if its ID is not included in the list.

Step 3: Scheduling intermediate nodes: In the third step of DASSA, each active node from tier *i* schedules NSD_i nodes from tier (*i* + 1) to be active where NSD stands for number of scheduled descendants. Active nodes from tier 1 schedule NSD_1 nodes from tier 2 with the highest energies to be active. Each selected tier 2 node then schedules NSD_2 nodes from tier 3 with the highest energies to be active, and so on. When a node which was scheduled to be active receives another scheduling message containing its node ID, it transmits a negative acknowledgement packet to the sender. Upon receiving this message, the sender schedules its neighbor with the next highest energy to be active. This

way, it is assured that NSD_i nodes will be scheduled by each active node from tier i . This provides a robustness to the algorithm but it was observed that it does not provide a substantial impact on the results.

Step 4: Transmitting and forwarding data: After the nodes decide their activity states, they forward their data together with their descendants' to one of their active neighbors from the upper tier with the highest energy. This is similar to the routing algorithm employed in [29]. In addition to the other steps in which nodes with higher residual energies are scheduled to be active, this step ensures that nodes with the maximum energies have higher loads. So, we apply a two-fold energy balancing scheme: both in activity scheduling and in routing. In fact, the purpose of this paper is not to design an efficient routing algorithm and any routing algorithm suitable for the purpose could be used together with the scheduling algorithm.

B. Communication overhead of DASSA: The control messages used in DASSA are neighbor discovery (Step 1) and scheduling (Step 2 and 3) messages. All nodes broadcast and receive neighbor discovery messages which are common in all sensor network algorithms in the literature [2]. The neighbor discovery messages in DASSA also include the residual energies of the nodes at the beginning of each round. Every round, nodes receive scheduling messages from their active neighbors closer to the sink and only nodes which are scheduled to be active transmit one broadcast message for scheduling their descendants. These scheduling messages contain the node IDs of a small number of nodes and hence are not very costly. This issue will be further explored in Sect. 5.

C. Selection of the parameters of DASSA: The parameters α and $\mathbf{NSD} = \{NSD_1, NSD_2, \dots, NSD_{N_{\text{tier}}-1}\}$, where N_{tier} is the number of tiers in the network, should be selected depending on GoC, R_s , R_t and field dimensions. This section provides an insight to the selection of these parameters assuming a uniform distribution for the sensors. DASSA parameters should be tuned further empirically to obtain the best results for a given network population and size using the results of this section as a basis. The results could be extended to any distribution as long as the integrations are tractable. Otherwise, the alternatives would be to stick with empirical methods or numerical integration.

Assume that a tier 1 node is at a distance r from the sink. The average number of tier 2 nodes in the communication range of this tier 1 node can be obtained by calculating the shaded area in Fig. 2 for the case when nodes are uniformly distributed. For given r , this shaded area is given by

$$S(r, \theta) = \pi R_t^2 - 2R_t^2 \arccos\left(\frac{r}{2R_t}\right) + \frac{1}{2}r\sqrt{4R_t^2 - r^2}.$$

After averaging this area over all r and θ , we obtain the average number of tier 2 neighbors of a tier 1 node, N_{12} , as follows,

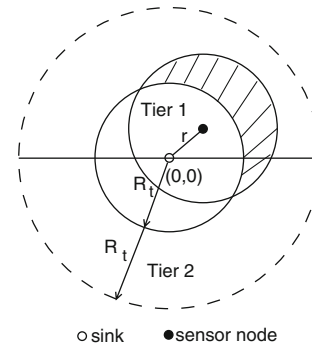


Fig. 2 Tier 2 nodes which are in the communication range of the tier 1 node are in the shaded area

$$\begin{aligned} N_{12} &= \left(\frac{N}{F}\right) \frac{2}{R_t^2} \int_0^{R_t} S(r, \theta) r dr \\ &= \frac{3\sqrt{3}}{4} R_t^2 \left(\frac{N}{F}\right), \end{aligned} \tag{14}$$

where F is the area of the sensor field. All tier 2 nodes are located in the area between the disk with radius R_t centered at the sink and the disk with radius $2R_t$ centered at the sink. Nodes from other tiers can also fall into this area but assuming a sufficiently dense network, this effect can be neglected. Using this approximation, the average number of tier 2 nodes, N_2 , can be approximated as

$$N_2 = 3\pi R_t^2 \left(\frac{N}{F}\right). \tag{15}$$

Let N_1 denote the number of active nodes from tier 1. Then N_1 should satisfy

$$N_1 \geq \alpha \frac{N_2}{N_{12}} = \frac{4\pi}{\sqrt{3}} \alpha \tag{16}$$

in order to be connected to α portion of the tier 2 nodes. The number of nodes, A , that are scheduled to be active by the algorithm is approximately given by

$$A = N_1 \left(1 + \sum_{i=1}^{N_{\text{tier}}} \prod_{j=1}^i NSD_j\right). \tag{17}$$

Combining (16) and (17), we obtain

$$A \geq \frac{4\pi}{\sqrt{3}} \alpha \left(1 + \sum_{i=1}^{N_{\text{tier}}} \prod_{j=1}^i NSD_j\right) \triangleq \underline{A}. \tag{18}$$

The minimum number of uniformly distributed nodes required to provide a certain GoC without considering connectivity is given by ([27], p. 39)

$$k = \left\lceil \frac{\log(1 - \text{GoC})}{DT} \right\rceil, \tag{19}$$

where the denominator (DT) is

$$DT = \log \left[\frac{1}{m^4} (n^2(m^2 - \pi R_s^2) + 4R_s m^2(n + R_s) - 4R_s^3(0.57 + 2.47n + 1.37R_s)) \right], \quad (20)$$

m is the length of the field and $n = m - 2R_s$.

The number of active nodes, A , should always be greater than this lower bound, i.e., $A \geq k$. Note that if $\underline{A} \geq k$, where \underline{A} is given by (18), then $A \geq k$. DASSA parameters can be selected such that $\underline{A} \geq k$ in order to satisfy GoC. In our simulations, α is generally chosen in the range 0.9–1 and **NSD** is computed so that $\underline{A} \geq k$.

Remarks k in (19) is the minimum number of uniformly distributed nodes required to satisfy a certain GoC without considering connectivity, whereas A is the number of connected and active nodes. We want to assure that there are at least k active sensors and thus we enforce $A \geq k$. Since connectivity is not considered in the derivation of this lower bound and k is independent of R_t , k is still a valid lower bound for satisfying GoC even when $R_s \neq R_t$. The tightness of this bound depends on the transmission and sensing radii and spatial distribution of sensor nodes. In the simulations reported in Sect. 5-D for unequal transmission and sensing ranges, i.e., $R_s \neq R_t$, this lower bound provided acceptable results.

Although the operation of DASSA does not require any specific channel model, the analysis discussed above for the selection of parameters uses the unit disc model. When DASSA is used in an environment for which a more realistic model, such as log-normal fading, is more appropriate, it may be recommended that DASSA parameters such as **NSD** and α are chosen larger than the values suggested by the above analysis so that more nodes will be activated and DASSA will have more robustness against channel impairments. This strategy can also be helpful in dealing with channel errors as each node will have more active descendants, compensating for the errored reports received from descendants.

5 Performance evaluation

This section provides comprehensive simulation results for evaluating the performances of CSSA, DASSA and DGT for various network populations and field sizes. We assume an ideal environment without any collisions and interference of packets since the focus of this work is to evaluate the scheduling algorithms on top layers rather than considering the physical layer issues. For finding the optimum solutions to ILP formulations, the CPLEX solver [30] is used in conjunction with the discrete event simulator built for the simulations.

A. Energy consumption model: The energy consumption model given in [9, 31] is used in the simulations. Each

sensor reports a 250-byte message to the sink at each round and each message transmission and reception consumes 0.1 mJ. The exchange messages used at the beginning of each round of DASSA are 25 bytes long and cost 10 μ J. Both transmitting and receiving a sleep control message which is 25 bytes long costs 10 μ J. A 20-byte header is included in each of these messages. The scheduling messages of DGT become larger as the desired GoC drops or as the number of nodes increases, and the transmission costs are increased accordingly.

B. Comparison of DASSA with other algorithms: Coverage plots of CSSA, DASSA and DGT for a 100 node network deployed on a 200 m-by-200 m sensor field with $E_{init} = 10$ mJ and $R_t = R_s = 50$ m are shown in Fig. 3. As GoC decreases, the gap between DASSA and CSSA decreases since the importance of exact location information decreases. Actually, for GoC = 0.8 and 0.7, the performance of DASSA is very close to CSSA. Also, it is important to note that with DASSA, the coverage remains almost all the time above GoC until it drops below it. However, in DGT, coverage is likely to fluctuate around the target due to the inherent probabilistic structure.

The performances of the algorithms for 4 different topologies each with 100 nodes uniformly deployed on a 200 m-by-200 m network field with $E_{init} = 10$ mJ are shown in Fig. 4. The results of DGT are averaged over 100 randomly generated runs. As expected, sleep scheduling algorithms provide a remarkable gain with respect to the case where all the nodes are active at all rounds. DASSA outperforms DGT in all the topologies. As GoC decreases, the gap between DASSA and CSSA decreases, whereas the gap between DASSA and DGT increases even more. The main reason is that DGT has to schedule redundant nodes in order to assure connectivity even though the GoC decreases. Also, DGT inhibits a random structure and cannot guarantee to provide the target GoC. As we can observe from Fig. 3, DGT provides excessive coverage in some rounds, which is above the desired GoC, while the coverage is below GoC for many rounds, and the network energy is wasted in those periods.

Figure 5 shows the performance of all algorithms for a 150 node network deployed in a 200 m-by-200 m field with $E_{init} = 10$ mJ. The relative performances of the algorithms in this setting are similar to the 100 node network cases.

C. Different network sizes: For a 200 m-by-200 m field with 200 nodes, 100 network deployments are simulated and the average performances of DASSA and DGT are compared for two different initial energy values of the nodes, $E_{init} = 10$ mJ and $E_{init} = 100$ mJ. The same set of parameters are used by each algorithm for all 100 sensor deployments. Figure 6 shows that DASSA achieves on the average 52–177% longer lifetimes compared with DGT.

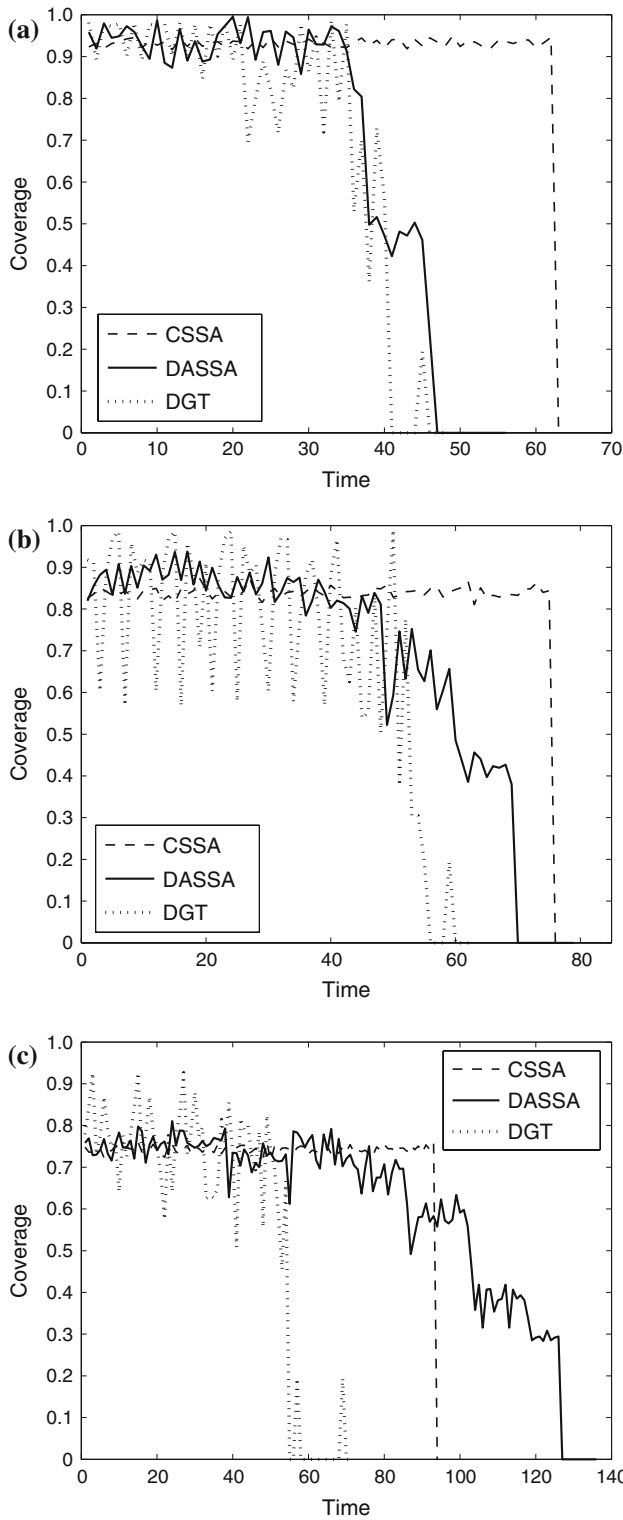


Fig. 3 CSSA, DASSA and DGT coverage plots for (a) GoC = 0.9, (b) GoC = 0.8, (c) GoC = 0.7

Similar conclusions can be drawn for $E_{init} = 10$ mJ and $E_{init} = 100$ mJ cases. The performance of CSSA could not be obtained for these larger networks due to excessive computation time.

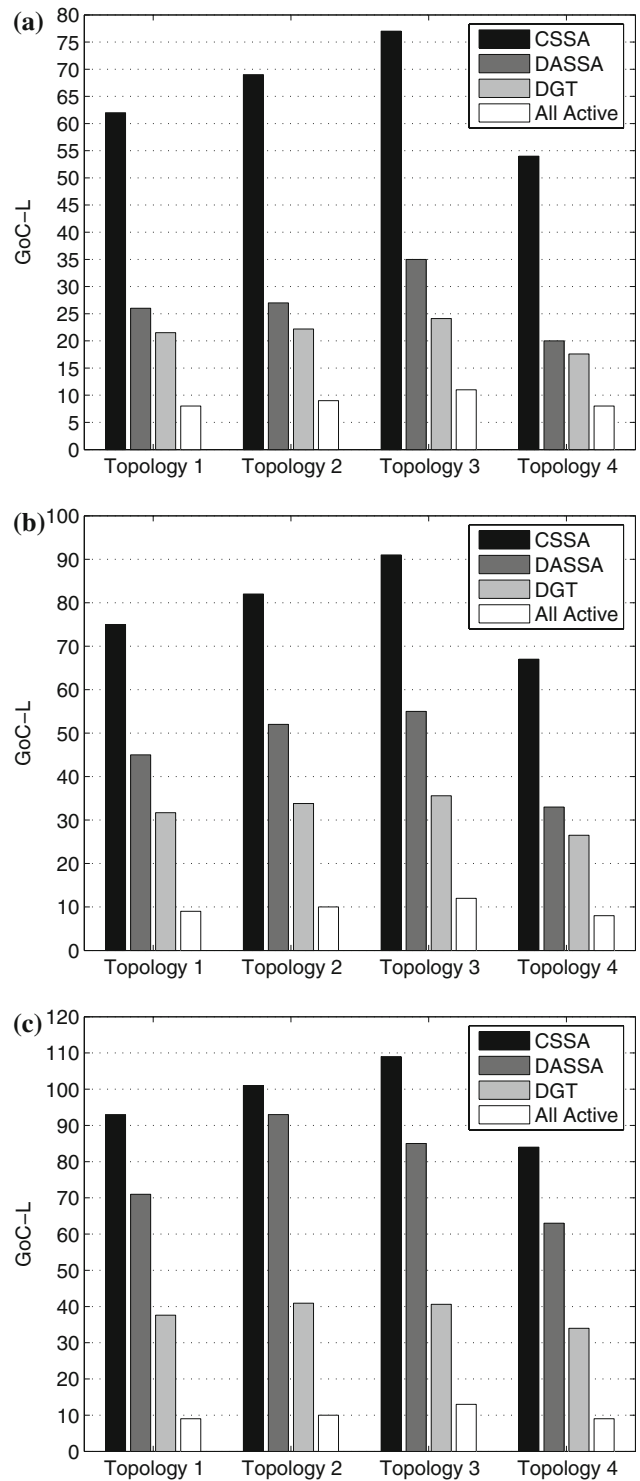


Fig. 4 GoC-L of all algorithms for 4 different topologies each with 100 nodes in a 200 m-by-200 m field. (a) GoC = 0.9, (b) GoC = 0.8, (c) GoC = 0.7

Finally, for a 300 m-by-300 m field with 400 nodes, 100 network deployments are simulated and the average performances of the distributed algorithms are compared for $E_{init} = 50$ mJ. Again, the same set of parameters are used

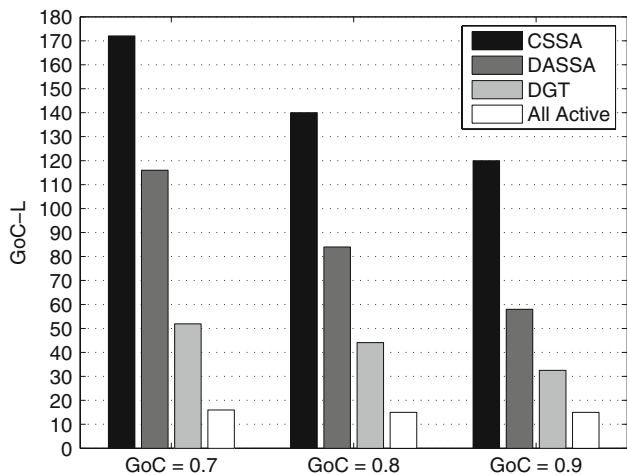


Fig. 5 Comparison of the algorithms for a 150 node network in a 200 m-by-200 m field

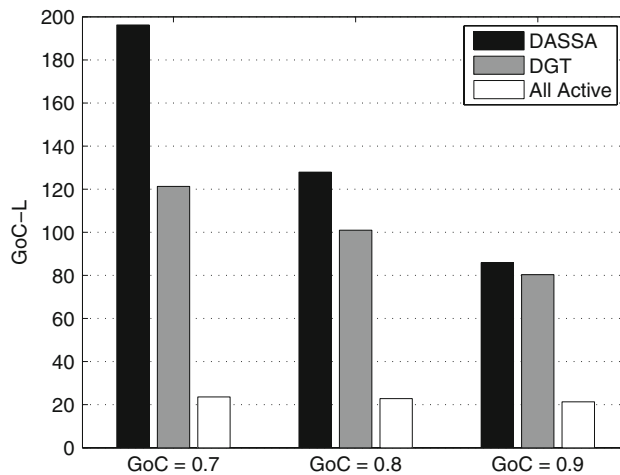


Fig. 7 Comparison of the algorithms for a 400 node network in a 300 m-by-300 m field

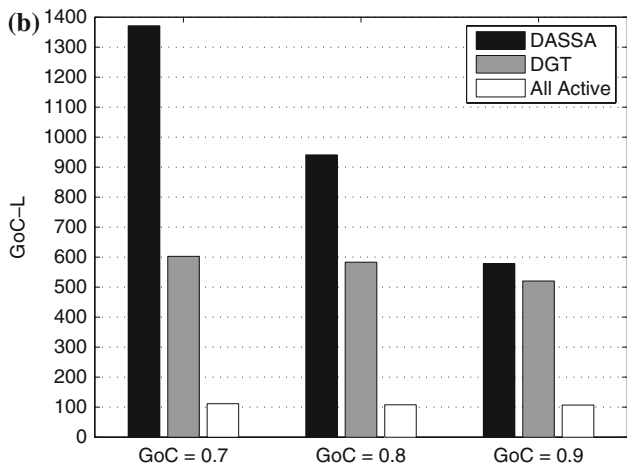
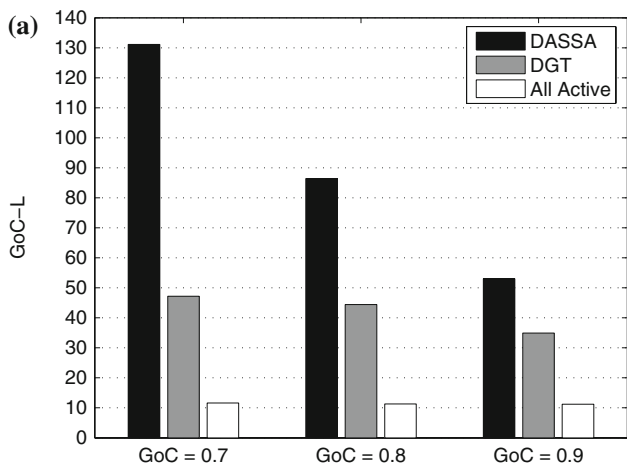


Fig. 6 Comparison of the algorithms for a 200 node network in a 200 m-by-200 m field. (a) $E_{init} = 10$ mJ, (b) $E_{init} = 100$ mJ

by each algorithm for all sensor deployments. Figure 7 shows that DASSA achieves on the average 7–62% longer lifetimes compared with DGT. As the field size increases,

the number of tiers increases. In DASSA, nodes in the first tier are selected in a balanced manner by using the feedback from the sink. However, as the number of tiers increases, the randomness in DASSA increases due to the lack of location information. Thus, DASSA starts to activate redundant nodes in order to assure sufficient coverage and because of this, its performance gain with respect to DGT decreases. As GoC decreases, the gap between DASSA and DGT increases and it is above 60% for $GoC = 0.7$.

D. Unequal transmission and sensing ranges: So far, it has been assumed that the sensing and transmission ranges are equal. We now analyze the effect of increasing the transmission range with respect to the sensing range: R_s is kept constant at 50 m and R_t is changed to 60 and 75 m when the network consists of 100 nodes, field is 200 m-by-200 m and $E_{init} = 10$ mJ. In DASSA, when the transmission range increases with respect to the sensing range, we only need to increase the number of nodes that a node activates from its next tier by increasing the values in **NSD**. Although less likely, if the transmission range becomes less than the sensing range, then DASSA should be adapted so that more nodes go into sleep by lowering **NSD** values.

We observe from Fig. 8 that DASSA achieves lifetimes close to CSSA, up to 90%, especially for lower GoC, and the relative performance of DASSA compared to CSSA is similar to the earlier cases. DGT could not be evaluated in this section since it does not address the case of unequal transmission and sensing ranges.

E. Effect of aggregation: In some applications, nodes might have the capability of combining the data packets before transmitting them [32]. We assume that each node combines all its descendants’ data into a single packet before transmitting, i.e., full aggregation. In Fig. 9, simulation results are shown for the same example considered in Fig. 4 with $E_{init} = 5$ mJ and for topology 1. We observe that

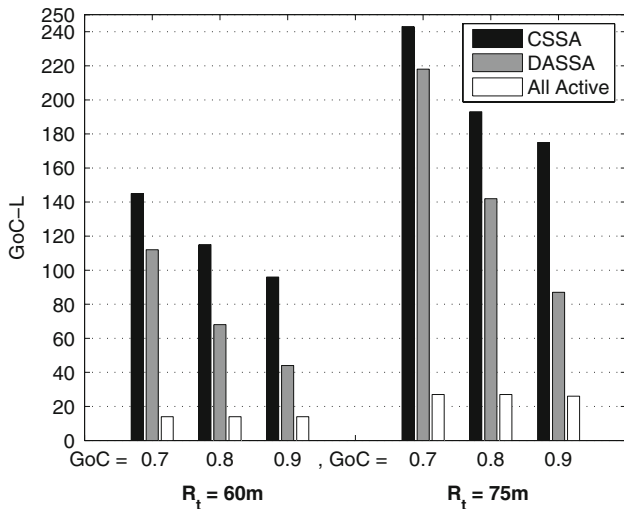


Fig. 8 Comparison of the algorithms for unequal R_t and R_s

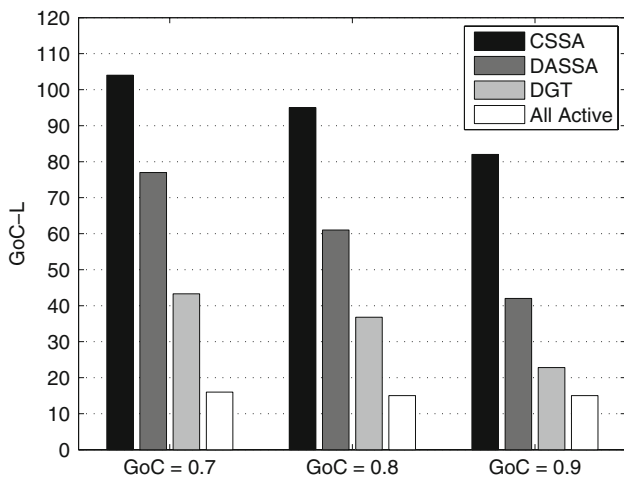


Fig. 9 Comparison of the algorithms with full aggregation for topology 1

DASSA still outperforms DGT by 65–80% in terms of GOC-L and aggregation does not have a significant effect on the relative performances of the algorithms. Note that CSSA requires some minor modifications when being used in the full aggregation case. Specifically, the variable f_{ij}^n in (1) and the expression in (4) has to be modified (see [27] for details).

F. Further analyses: The performance of DASSA is further studied by observing the evolution of the total energy and the number of dead and connected nodes in the network. Figure 10 shows the total residual energy of all the nodes and the number of nodes which are active and dead for $GoC = 0.8$ as the network evolves. CSSA consumes about the same amount of energy at each round in a perfectly balanced manner. DASSA consumes energy similar to CSSA but much more efficiently than DGT. In CSSA, all the nodes start to die approximately at the same time due to the balanced energy consumption. Figure 10(a) shows that more

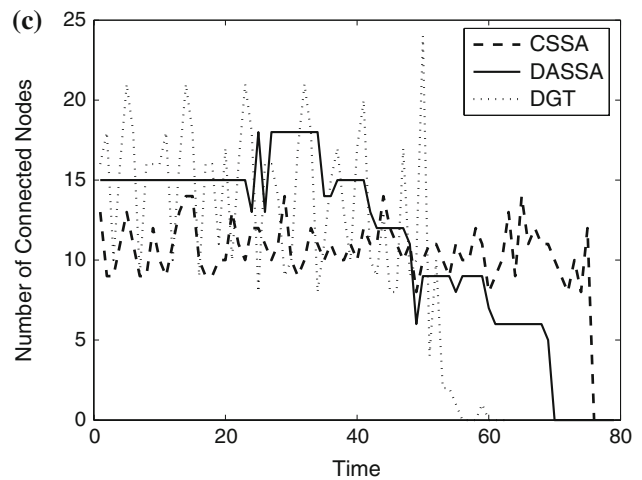
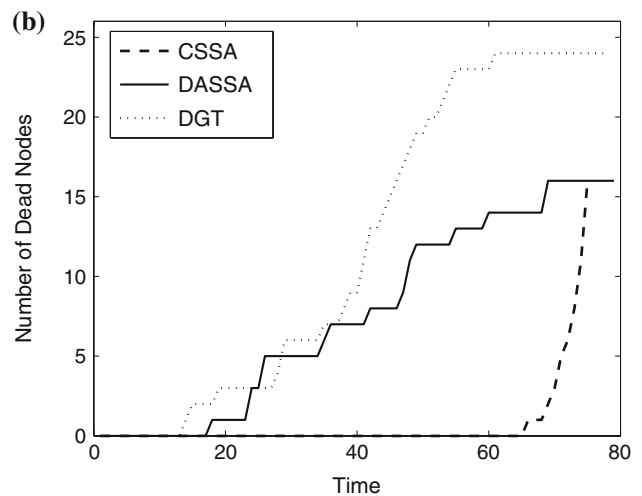
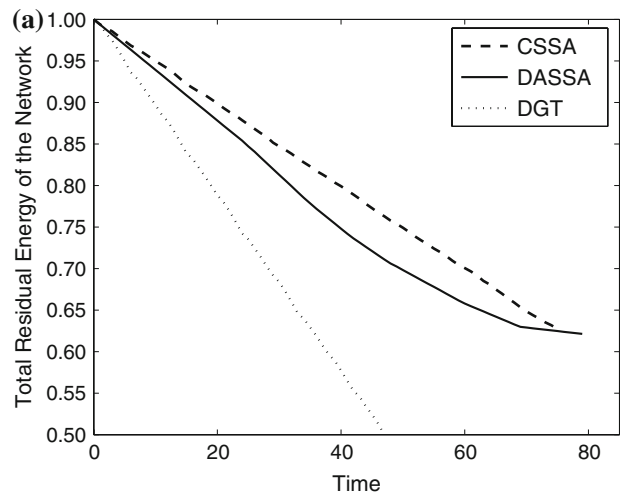


Fig. 10 Energy and scheduling plots for $GoC = 0.8$ for the first topology; (a) Normalized total residual energy of all nodes with respect to 1 J. (b) Number of nodes which are dead at each round. (c) Number of nodes connected to the sink at each round

than 50% of the total energy of the network remains when the network operation ends since all tier 1 nodes die. This once more emphasizes the importance of the scheduling of tier 1

nodes. Figure 10(c) shows that approximately same number of nodes are activated at each round by CSSA whereas DASSA and DGT activate larger number of sensors compared with CSSA. The number of nodes activated by DGT fluctuates due to severe randomness.

Figure 11 shows the number of rounds each location in the field is covered, i.e., the coverage count of each location, by all

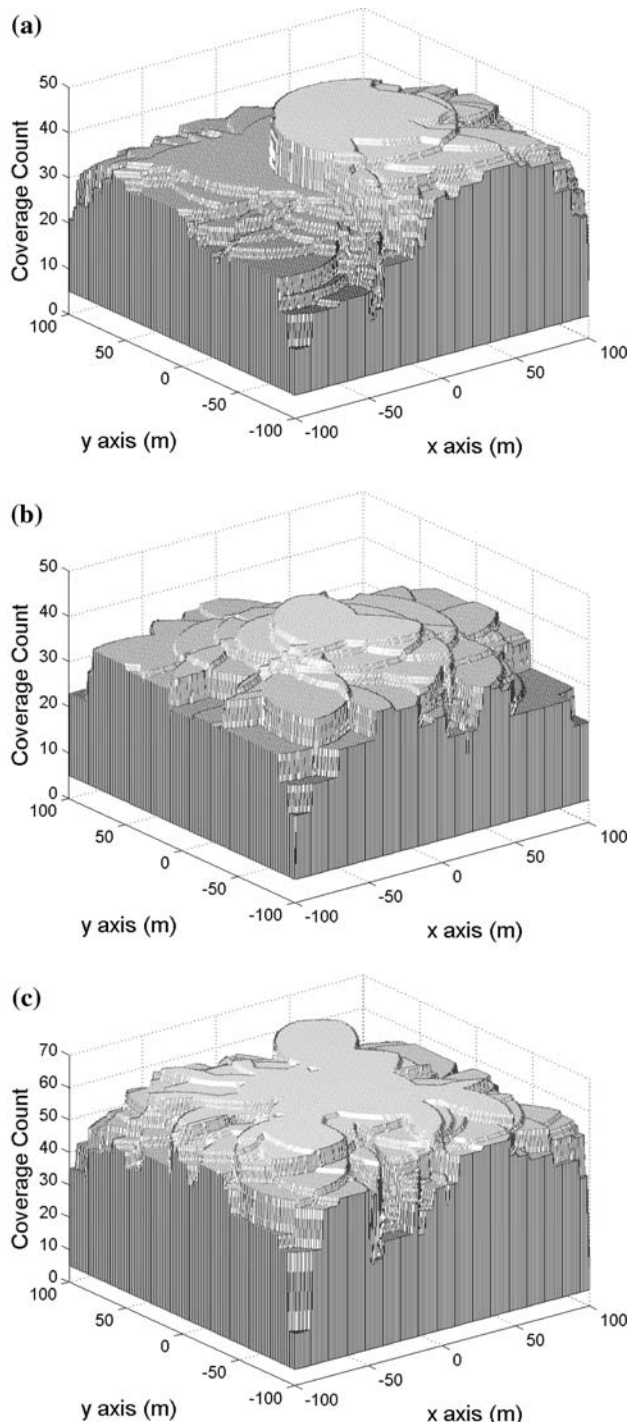


Fig. 11 Coverage counts for GoC = 0.9 for topology 1. (a) DASSA, (b) DGT, (c) CSSA

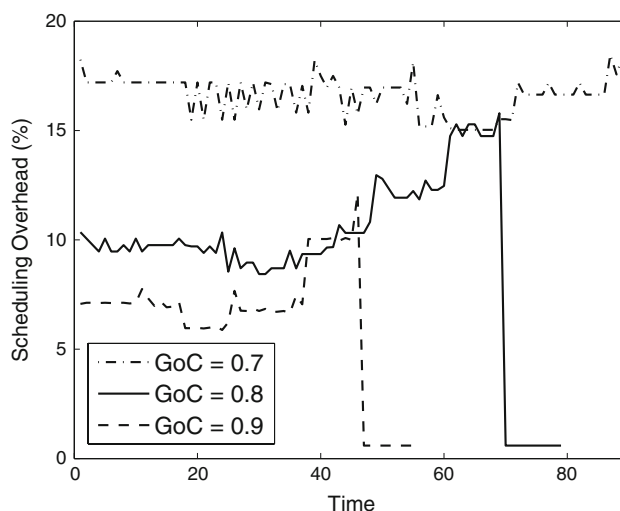


Fig. 12 Scheduling overhead of DASSA for topology 1. Percentages are with respect to the total consumed energy at each round

four algorithms for GoC = 0.9. CSSA balances the coverage in the field even though the objective is to maximize the number of rounds the coverage is above GoC and no constraint for balanced coverage is introduced in the ILP formulation. Similarly, the aim of DASSA is to maximize the number of rounds for which the coverage is above GoC without any concern in balancing the coverage. However, DASSA maintains a balanced coverage at each location and achieves far better coverage counts than DGT, but not as balanced as CSSA especially near the edges of the sensor field.

Finally, Fig. 12 shows the percentage of the scheduling overhead for DASSA at each round for Topology 1, which is used for the results shown in Fig. 3. We observe that the scheduling overhead of DASSA is around 5–20% as compared to the total energy consumption. When GoC decreases, the energy consumption decreases faster than the decrease in the overhead, which increases the relative energy spent for scheduling. Considering that the cost of transmitting a scheduling message corresponds to 10% of the cost of transmitting a data packet, these levels are quite acceptable. The jumps in the plots correspond to the times where DASSA cannot sustain the target GoC anymore.

6 Conclusion

In this paper, the activity scheduling problem in wireless sensor networks with partial coverage is investigated. The optimum scheduling of nodes is found by using an ILP formulation which provides the maximum number of rounds for which the network can satisfy a certain coverage level. This approach is centralized and requires location information and global knowledge of the network. Next, a

distributed, simple and scalable sleep scheduling algorithm called DASSA is proposed. The main objective of DASSA is to find the minimum set of nodes which can satisfy the desired coverage without using any location information and only using local information. An analysis for selecting the parameters of DASSA is also provided. DASSA can attain network lifetimes up to 46, 63 and 92% of the lifetimes achieved by the centralized algorithm for coverage levels of 90, 80 and 70%, respectively. DASSA is also shown to significantly outperform DGT sleep scheduling algorithm in terms of the network lifetime subject to partial coverage constraints.

As the number of tiers in the network increases, the gain provided by DASSA decreases since the randomness in DASSA increases. In such sensor networking applications, DASSA can benefit from a clustered network structure. A multi-sink scenario where each sink employs DASSA could increase the performance of the algorithm. DASSA can also be used in an heterogeneous network, where a subset of nodes have higher capabilities than the other type of nodes. In such a network, high powered nodes can employ DASSA within their clusters. For future work, the performance of DASSA can be analyzed for a clustered network.

Acknowledgement This research has been conducted within the NEWCOM++ Network of Excellence in Wireless Communications funded through the EC 7th Framework Programme.

References

1. Akyildiz, I. F., Su, W., Sankarasubramanian, Y., & Cayirci, E. (2005). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.
2. Wang, L., & Xiao, Y. (2006). A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5), 723–740.
3. Xu, Y., Heidemann, J., & Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing. In *Proceedings of ACM MOBICOM*.
4. Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2001). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Mobile Computing and Networking*, pp. 85–96.
5. Cerpa, A., & Estrin, D. (2002). ASCENT: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM*, June 2002.
6. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, pp. 1567–1576.
7. Ye, F., Zhong, G., Cheng, J., Lu, S., & Zhang, L. (2003). PEAS: A robust energy conserving protocol for long-lived sensor networks. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, Providence, Rhode Island, USA, p. 28.
8. Meguerdichian, S., & Potkonjak, M. (2003). Low power 0/1 coverage and scheduling techniques in sensor networks. University of California Los Angeles, Technical Reports 030001, January 2003.
9. Tian, D., & Georganas, N. (2002). A coverage-preserving node scheduling scheme for large wireless sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*.
10. Tian, D., & Georganas, N. (2003). A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing*, 3(2), 271–290.
11. Boukerche, A., Fei, X., Araujo, R. B., & Patnaik, P. (2006). A local information exchange based coverage-preserving protocol for wireless sensor networks. In *Proceedings of IEEE ICC*.
12. Gupta, H., Das, S., & Gu, Q. (2003). Connected sensor cover: Self organization of sensor networks for efficient query execution. In *ACM Mobile Adhoc Network Symposium (MOBIHOC)*, pp. 189–199.
13. Cardéi, M., & Du, D.-Z. (2005). Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3), 333–340.
14. Wang, X., Zhang, Q., Sun, W., Wang, W., & Shi, B. (2006). A coverage-based maximum lifetime data gathering algorithm in sensor networks. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)* (p. 33). Washington, DC, USA: IEEE Computer Society.
15. Wang, X. et al. (2003). Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of Sensys*.
16. Zhang, H., & Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *International Journal of Wireless Ad Hoc and Sensor Networks*, 1(12), 89–124.
17. Liu, Y., & Liang, W. (2005). Approximate coverage in wireless sensor networks. In *Proceedings of 30th Annual IEEE Conference on Local Computer Networks, IEEE Computer Society*, November 2005, pp. 68–75.
18. Xu, Y., Heidemann, J., & Estrin, D. (2005). pCover: Partial coverage for long-lived surveillance sensor networks. Department of Computer Science, Michigan State University, Tech. Rep., November 2005.
19. Lu, J., & Suda, T. (2003). Coverage-aware self-scheduling in sensor networks. In *IEEE 18th Annual Workshop on Computer Communications*, October 2003, pp. 117–123.
20. Zhang, H., & Hou, J. (2004). On deriving the upper bound of α -lifetime for large sensor networks. In *MobiHoc '04* (pp. 121–132). New York, NY, USA: ACM Press.
21. Wu, K., Gao, Y., Li, F., & Xiao, Y. (2005). Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Kluwer MONET Journal, Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, 10(6), 837–852.
22. Choi, W., & Das, S. K. (2005). A novel framework for energy-conserving data gathering in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, March 2005.
23. Yang, X., & Vaidya, N. H. (2004). A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay. In *RTAS '04: 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 19–26.
24. Elson, J., & Estrin, D. (2001). Time synchronization for wireless sensor networks. In *15th International Parallel & Distributed Processing Symposium (IPDPS)*.
25. Ganeriwal, S., Kumar, R., & Srivastava, M. B. (2003). Timing-sync protocol for sensor networks. In *ACM Conference on Embedded Networked Sensor Systems (SENSYS)*.
26. Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. J. (2003). Energy-efficient collision-free medium access control for wireless sensor networks. In *SensSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 181–192.
27. Yardibi, T. (2006). Sleep scheduling for energy conservation in wireless sensor networks with partial coverage. Master's thesis,

Bilkent University, Ankara. [Online]. Available: <http://www.ee.bilkent.edu.tr/ytarik/thesis.pdf>

28. Hohlt, B., Doherty, L., & Brewer, E. (2004). Flexible power scheduling for sensor networks. In *IPSN '04: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pp. 205–214.
29. Leuschner, C. J. (2005). The design of a simple energy efficient routing protocol to improve wireless sensor network lifetime. MS thesis. Faculty of Engineering, University of Pretoria, April 2005.
30. ILOG CPLEX 10.0. [Online]. Available: <http://www.ilog.com/products/cplex/>
31. Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00* (p. 8020). Washington, DC, USA: IEEE Computer Society.
32. Boulis, A., Ganeriwal, S., & Srivastava, M. B. (2003). Aggregation in sensor networks: An energy-accuracy trade-off. *Elsevier Ad Hoc Networks*, 1(23), 317–331.

Author Biographies



Tarik Yardibi received the B.S. degree from Hacettepe University, Ankara, Turkey in 2004 and the M.S. degree from Bilkent University, Ankara, Turkey in 2006, both in Electrical Engineering. He is currently pursuing the Ph.D. degree at the Spectral Analysis Laboratory at University of Florida, Gainesville. His research interests include wireless sensor networks, statistical signal processing, sensor array processing and sparse representations.



Ezhan Karasan received B.S. degree from Middle East Technical University, Ankara, Turkey, M.S. degree from Bilkent University, Ankara, Turkey, and Ph.D. degree from Rutgers University, Piscataway, New Jersey, USA, all in Electrical Engineering, in 1987, 1990, and 1995, respectively. During 1995–1996, he was a post-doctorate researcher at Bell Labs, Holmdel, New Jersey, USA. From 1996 to 1998, he was a

Senior Technical Staff Member in the Lightwave Networks Research Department at AT&T Labs-Research, Red Bank, New Jersey, USA. He has been with the Department of Electrical and Electronics Engineering at Bilkent University since 1998, where he is currently an associate professor. During 1995–1998, he worked in the Long Distance Architecture task of the Multiwavelength Optical Networking Project (MONET), sponsored by DARPA. Dr. Karasan is a member of the Editorial Board of Optical Switching and Networking journal. He is the recipient of 2004 Young Scientist Award from Turkish Scientific and Technical Research Council (TUBITAK), 2005 Young Scientist Award from Mustafa Parlar Foundation and Career Grant from TUBITAK in 2004. Dr. Karasan received a fellowship from NATO Science Scholarship Program for overseas studies in 1991–1994. His current research interests are in the application of optimization and performance analysis tools for the design, engineering and analysis of optical networks and wireless ad hoc/sensor networks.