Dynamic Sensing Scheduling to Prolong Network Lifetime under Practical Requirements

Yang Peng
University of Washington Bothell
Bothell, Washington 98011, USA
Email: yangpeng@uw.edu

Zi Li, Wensheng Zhang and Daji Qiao
Iowa State University
Ames, Iowa 50011, USA
Email: {zili, wzhang, daji}@iastate.edu

***Abstract*—We propose a unique Dynamic Sensing Scheduling (DSS) scheme to prolong the lifetime of a sensor network. Different from most existing works, we study the sensor network lifetime under two practical requirements: *sensing coverage* and *network connectivity*. A sensor node is considered *critical* if its depletion of energy would cause either a violation of the sensing coverage requirement (specified by the application) or a disconnection of the routing tree. The key idea of DSS is to adjust the sensing duties of sensor nodes according to their nodal lifetime as well as their criticality. Under this design principle, DSS schedules more sensing duties to non-critical nodes (even at the cost of losing them more quickly) so that critical nodes may stay alive for a longer period of time, thus extending the network lifetime. DSS adjusts the sensing duties between neighboring nodes only, and is a distributed and lightweight solution. Simulation results show that DSS performs well under various network setups, close to a theoretical upper bound.**

I. Introduction

When a wireless sensor network is deployed for long-term continuous monitoring, it is essential to prolong its lifetime as much as possible. Various definitions of network lifetime have been proposed and studied in the literature [7], [8], [9], [19], [24], [26], [27]. Among them, a simple but widely adopted definition of network lifetime is *the earliest time when any node in the network dies, i.e., the minimal nodal lifetime* [8], [9]. Many schemes [3], [4], [13], [14], [15], [17] have been proposed to extend the network lifetime in terms of this definition. Balancing nodal residual energy, lifetime, or energy consumption rates are the techniques commonly used in these schemes.

However, such a simple definition of network lifetime may be unrealistic because sensor nodes are usually deployed with a high level of redundancy in practice. In the presence of node redundancy, the network may still produce satisfactory monitoring results even when a certain number of nodes have died earlier than others. A more practical definition of network lifetime could be *the earliest time when a certain level of application-required sensing coverage cannot be satisfied or any sensor node with an assigned sensing duty does not have a path to forward its sensory data to the sink* [7], [24], [27]. Under this definition, the network lifetime may not be the minimal nodal lifetime any more. For example, in the simple network shown in Figure 1(a), if the application only requires at least one node to perform sensing tasks at any time, the network lifetime simply becomes the nodal lifetime of node 1. In another example network shown in Figure 1(b), the network lifetime becomes the maximal nodal lifetime between nodes 1 and 2, because as long as one of them remains alive and performs sensing tasks, the required sensing coverage is guaranteed. Few works have been reported to study how to extend the network lifetime under this more practical definition, which is the focus of this paper.

Fig. 1. Example networks.

In this paper, we propose a unique DSS (Dynamic Sensing Scheduling) scheme to prolong the sensor network lifetime. DSS is a distributed and lightweight solution, where the sensing duty adjustment is triggered only when communication occurs between neighboring nodes, and the sensing duties are adjusted locally between a parent node and its child nodes.

In DSS, sensor nodes are classified as *critical* and *non-critical* nodes, where critical nodes are referred to as the nodes whose depletion of energy would render the sensor network non-functional as either or both of the practical sensing coverage and network connectivity requirements cannot be satisfied. Thus, the network lifetime becomes the minimal nodal lifetime of critical nodes. DSS attempts to shift the sensing duties from critical to non-critical nodes as much as possible, while between critical nodes, longer-lifetime ones tend to take more sensing duties to balance the expected lifetime of critical nodes. In case all nodes are non-critical, more sensing duties are assigned to shorter-lifetime ones instead; this way, the overall idling time of non-critical nodes is reduced, meaning that the non-critical nodes use their energy more efficiently and thus the network lifetime may be extended (indirectly).

Extensive simulations have been conducted to evaluate the performance of the proposed DSS scheme. Results show that, DSS outperforms the compared schemes, and yields a performance close to a derived theoretical upper bound.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the system model, the problem statement, and an analysis of the performance upper bound. Details of the DSS design are elaborated

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

in Section IV. Simulation results are reported in Section V. Finally, Section VI concludes the paper.

II. Related Work

Under different definitions of sensor network lifetime, a large variety of techniques have been proposed to prolong the network lifetime. In this section, we first summarize the related work on the network lifetime definition and then the lifetime extension techniques.

1) Definitions of network lifetime: The authors of [7] presented a comprehensive survey on the definitions of sensor network lifetime. The most widely-used one defines the network lifetime as the time when the first sensor node runs out of energy [8], [9], which assumes all nodes in the network are equally important. On the other hand, if the sensing coverage is the main concern, the network lifetime can be defined as the time when a monitored target or area cannot be sensed with a certain required fidelity, such as k-coverage [24] and α-coverage [26]. Similarly, if the network connectivity is the main concern, the network lifetime can also be defined as the time when either the network connectivity or the coverage ratio drops below a certain threshold [19], [27]. The network lifetime definition adopted in this paper is based on both sensing coverage and network connectivity requirements.

2) Techniques to prolong network lifetime: Numerous schemes have been proposed to prolong the lifetime of a sensor network. Among them, node sleep/active scheduling schemes [1], [2], [19], [24] turn on/off nodal sensing and/or communication activities to save energy while achieving application desired sensing qualities in sensor networks with redundant nodes. Energy-aware routing protocols [10], [13], [16], [20] route packets through nodes with a higher residual energy or a longer nodal lifetime such that nodes with a lower residual energy or a shorter nodal lifetime can live longer by participating less in data forwarding. MAC layer techniques [3], [14], [17], [28] dynamically tune parameters such as channel checking period and data retransmission interval, under application-specified constraints. The goal is to adjust the distribution of communication overhead between different nodes so that the network lifetime can be extended. In addition, cross-layer approaches [4], [5], [6], [12], [15], [21] have also been proposed. For example, [6] attempts to maximize the network lifetime via joint routing and congestion control. Recently, two new cross-layer protocols were proposed in [15], [21]: I^2C – a joint routing and MAC protocol, and JAM – a joint data aggregation and MAC protocol, which enable neighboring nodes to collaborate locally to extend the lifetime of duty cycle sensor networks. All the afore-discussed schemes were proposed for networks in which all nodes play an equally important role and network lifetime is defined as the time when the first node dies. Hence, an essential component in these schemes is to balance nodal residual energy, lifetime, or energy consumption rates between sensor nodes.

By defining the network lifetime according to the sensing coverage, the authors of [26] studied how to deploy sensor nodes to reduce network energy consumption and provide desired sensing coverage. The authors of [11], [19] proposed to maximize the network lifetime via changing sleep/active status of sensor nodes. However, the sensing duties of neighboring nodes are not jointly adjusted in these schemes.

TABLE I
STRATEGIES TO PROLONG THE NETWORK LIFETIME

network lifetime definition	related work	techniques
minimal nodal lifetime	[10], [13], [16], [20]	energy-aware routing
	[3], [14], [17], [28]	energy-aware MAC
	[4], [5], [6], [12]	cross-layer
	[8]	clustering
according to sensing coverage requirement	[26]	planned deployment
	[24]	node sleep/active scheduling
according to both sensing coverage and connectivity requirements	[1], [2], [11], [19]	node sleep/active scheduling
	DSS	dynamic sensing scheduling between neighboring nodes

Table I summarizes the existing techniques to prolong the network lifetime under different network lifetime definitions. The DSS scheme proposed in this paper is designed with the awareness of redundant nodes available in the network, and under the more realistic network lifetime definition as the time when either a required level of sensing coverage or network connectivity fails.

III. Model and Problem Statement

A. System Model

We consider a sensor network with one sink and N sensor nodes. The network is deployed to monitor M non-overlapping areas. We assume that each node knows the area that it resides in. In each area ℓ ($\ell = 1, \cdots, M$), there are n_ℓ nodes, and we denote them as nodes $1, 2, \cdots, n_\ell$ without loss of generality.

We study the following sensing coverage requirement, where each area ℓ shall be α_ℓ-covered; that is,

$$\sum_{i=1}^{n_\ell} S_i \geqslant \alpha_\ell, \tag{1}$$

where S_i ($0 \leqslant S_i \leqslant 1$) is the sensing duty cycle of node i ($i = 1, 2, \cdots, n_\ell$). This means that node i shall be actively sensing for S_j time every single time unit; the total number of sensing samples collected per time unit shall be at least $\alpha_\ell \beta$, where β is the number of sensing samples generated per time unit by a sensor node with 100% sensing duty cycle. In order to deliver sensory data to the sink, all the alive nodes in network shall form a routing tree rooted at the sink. We assume that aggregation is not performed by forwarding nodes in this work. The network lifetime is defined as *the earliest time when either the sensing coverage requirement is violated in any monitoring area, or any sensor node i with $S_i > 0$ is disconnected from the sink.*

B. Problem Statement

Formally, the problem studied in this paper can be described in a time-discrete manner as follows. Notations used in this paper are summarized in Table II.

TABLE II
SUMMARY OF NOTATIONS

notation	meaning
α_ℓ	sensing coverage requirement of area ℓ
β	number of sensing samples generated per time unit by a sensor node with 100% sensing duty cycle
θ_{rx}	energy consumed to receive one sensing sample
θ_{tx}	energy consumed to transmit one sensing sample
θ_s	energy consumed to collect one sensing sample
θ_ϵ	energy consumed per time unit when a node is idle
$f_{i\to j}$	number of sensing samples transmitted from i to j
S_i	sensing duty cycle of node i
$\mathcal{C}_i$	set of child nodes of node i
$\mathcal{P}_i$	set of parent nodes of node i
e_i	residual energy of node i
c_i	energy consumption rate of node i
L_i	lifetime of node i

Objective:

- $\max\{T\}$

Given:

- θ_{rx}, θ_{tx}, θ_s, θ_ϵ, and β
- For each area ℓ: α_ℓ
- For each node i: e_i, $\mathcal{C}_i$, and $\mathcal{P}_i$

Subject to: $\forall t \in \{0, \cdots, T\}$:

- *Sensing Coverage Requirement:*

$$\sum_{i=1}^{n_\ell} S_i(t) \geqslant \alpha_\ell \text{ and } 1 \geqslant S_i(t) \geqslant 0$$

- *Connectivity Requirement:*

$$e_i(t) \geqslant S_i(t)\beta\theta_s + \sum_{j\in\mathcal{P}_i} f_{i\to j}(t)\theta_{tx} + \sum_{k\in\mathcal{C}_i} f_{k\to i}(t)\theta_{rx} + \theta_\epsilon$$

- *Network Flow Requirement:*

$$\sum_{j\in\mathcal{P}_i} f_{i\to j}(t) = \sum_{k\in\mathcal{C}_i} f_{k\to i}(t) + S_i(t)\beta$$

Output:

- For each node i, its sensing duty $S_i(t)$, $t \in \{0, \dots, T\}$.

Directly solving the above problem in a centralized manner may be impractical in real-world sensor networks, as a large amount of information about each sensor node needs to be collected and refreshed frequently, and distribution of decisions to individual nodes may incur high communication overhead. Hence, we develop a distributed scheme to solve the problem, and compare its performance with a theoretical upper bound, which is derived next.

C. Analysis of Performance Upper Bound

The connectivity requirement can be relaxed by neglecting θ_ϵ as follows:

$$e_i(t) \geqslant S_i(t)\beta\theta_s + \sum_{j\in\mathcal{P}_j} f_{i\to j}(t)\theta_{tx} + \sum_{k\in\mathcal{C}_i} f_{k\to i}(t)\theta_{rx}.$$

Then, an upper bound on T may be obtained by using a non-linear programming (NLP) solver. However, as the network size increases, the number of variables may become too large for the solver to output a solution within a reasonable period of time. Therefore, to further speed up the computation, we instead apply the NLP solver to the following amortized version of the problem, where $\overline{S}_i = (\sum_{t=0}^{T-1} S_i(t))/T$, $\overline{f}_{i\to j} = (\sum_{t=0}^{T-1} f_{i\to j}(t))/T$, and $\overline{f}_{k\to i} = (\sum_{t=0}^{T-1} f_{k\to i}(t))/T$.

Objective:

- $\max\{T\}$

Given:

- θ_{rx}, θ_{tx}, θ_s, θ_ϵ, and β
- For each area ℓ: α_ℓ
- For each node i: e_i, $\mathcal{C}_i$, and $\mathcal{P}_i$

Subject to:

- $\sum_{i=1}^{n_\ell} \overline{S}_i \geqslant \alpha_\ell$, $1 \geqslant \overline{S}_i \geqslant 0$
- $e_i \geqslant (\overline{S}_i\beta\theta_s + \sum_{j\in\mathcal{P}_i} \overline{f}_{i\to j}\theta_{tx} + \sum_{k\in\mathcal{C}_i} \overline{f}_{k\to i}\theta_{rx}) \cdot T$
- $\sum_{j\in\mathcal{P}_i} \overline{f}_{i\to j} = \sum_{k\in\mathcal{C}_i} \overline{f}_{k\to i} + \overline{S}_i\beta$

Output:

- For each node i, its averaged sensing duty $\overline{S}_i$.

Now, we have reduced the numbers of variables and constraints significantly, and this amortized version can be solved using an NLP solver easily. On the other hand, notice that any solution to the original problem is also a solution to this amortized version, as the constraints are being relaxed. Therefore, the solution to this amortized version upper bounds the solution to the original problem. The obtained upper bound will be used for comparison when we evaluate the performance of DSS in Section V.

IV. DSS DESIGN

In this section, we present DSS (Dynamic Sensing Scheduling), a distributed and lightweight solution that adjusts sensing activities dynamically between neighboring nodes in a sensor network, so that the network lifetime defined in Section III-A can be prolonged. DSS works as follows.

- Initially, the entire field is partitioned into multiple monitoring areas determined by the application. When the partitioning changes over time, the sink node will broadcast a message with "Area ID" and "Sensing Coverage Requirement" to nodes in the affected areas.
- Then, a routing tree rooted at the sink is constructed to connect all nodes for sensory data collection, and nodes are assigned with proper sensing duties to ensure that the sensing coverage requirement in every monitoring area is satisfied. The tree may be constructed using well-known energy-aware routing protocols [13], [20].
- After initialization, the sensing duty assignment is adjusted continuously in a local and gradual manner throughout the network lifetime. The purpose of the adjustment is to adapt the sensing activities to the changes

in system conditions (e.g., distribution of nodal residual energy and lifetime), so as to extend the network lifetime as much as possible. Also, as the adjustment is only performed between neighboring nodes, the incurred communication overhead is low.

A. Definition of Critical Nodes

DSS starts with the identification of *critical nodes* in the network, which are defined as the nodes whose depletion of energy would cause either a violation of the sensing coverage requirement or a disconnection of any other sensor nodes (with non-zero sensing duty) from the routing tree. In other words, the death of a critical node would render the whole sensor network nonfunctional. Hence, the network lifetime is the minimal nodal lifetime of critical nodes.

Figure 2 shows an example network that consists of four monitoring areas and each area requires 2-sensing coverage. In this network, nodes 1, 2, 3, 5, 8, 10, 13, and 14 are critical nodes according to the definition, as the death of any of them would render the sensor network nonfunctional. For example, if node 14 dies, there is only one active sensor in its monitoring area, and hence the 2-sensing coverage requirement cannot be satisfied. In another example, if node 8 dies, nodes 10 to 12 are disconnected from the routing tree; thus, the sensed data cannot be reported to the sink, meaning that their residing area is not monitored properly. On the other hand, when either node 11 (or 12) dies, the 2-sensing coverage requirement can still be satisfied with node 12 (or 11) and the critical node 10. Therefore, neither node 11 nor 12 is considered a critical node in this example.

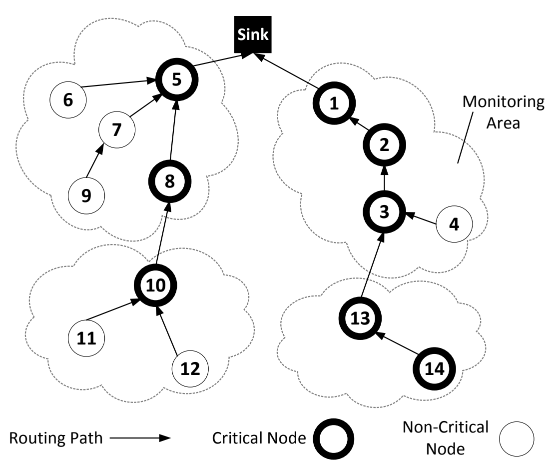


Fig. 2. An example network with critical nodes identified with bold circles. There are four monitoring areas in the network, and the 2-sensing coverage is required for each area.

B. Identification of Critical Nodes

For each node to decide whether it is a critical node, the node needs to collect the following information: (1) the sensing coverage requirement of its residing area; (2) the total number of sensor nodes deployed in this area; and (3) the number of nodes that belong to the node's subtree and also reside in the same area. The first two pieces of information can be embedded into the notification messages when establishing/updating monitoring areas. The third piece of information can be obtained by asking each node to embed its area information (e.g., area ID) in data packets, and monitor this information when forwarding data packets. With these information, each node can decide whether it is a critical node, with limited communication overhead.

DSS attempts to extend the network lifetime with the following strategy: (1) prolong the minimal nodal lifetime of critical nodes via shifting workloads from critical nodes to non-critical nodes, or from shorter-lifetime critical nodes to longer-lifetime critical nodes; (2) improve the energy efficiency of non-critical nodes. Note that the latter approach may not extend the network lifetime directly, as the minimal nodal lifetime of critical nodes is not extended immediately. However, as it may extend the overall lifetime of all non-critical nodes, it may delay the moment when the non-critical nodes die and their workloads have to be shifted to critical nodes; this way, the network lifetime is extended indirectly.

C. Parent Node Behavior

Every time when communication occurs between a pair of parent and child nodes, the parent node checks whether the sensing duties assigned to itself and its child nodes can be adjusted to extend the network lifetime.

More specifically, upon receiving a data packet from its child node i, the parent node extracts node i's lifetime L_i. With the knowledge of nodal lifetime and criticality of each child node i, the parent node adjusts the sensing duties of itself and its child nodes as follows.

- Case 1: All nodes are critical. In this case, all nodes are equally important for the sensing coverage requirement. DSS employs a lifetime balancing strategy by *shifting the sensing duties from the shortest-lifetime critical node to the longest-lifetime critical node*.
- Case 2: All nodes are non-critical. In this case, the sensing coverage won't be affected when any of the nodes runs out of energy. According to the design principle, the goal of adjustment becomes to improve the energy efficiency of non-critical nodes. Therefore, DSS *shifts the sensing duties from the parent node or the longest-lifetime child node to the shortest-lifetime child node*.
- Case 3: There exist both critical and non-critical nodes. This case combines the previous two cases, and DSS *shifts the sensing duties from the shortest-lifetime critical node to the shortest-lifetime non-critical node*.

In both Cases 2 and 3, in order to improve the energy efficiency of non-critical nodes, the *shortest-lifetime child node* is selected to perform more sensing duties. The reasons for this are two-fold. First, shifting sensing duties from parent to child node can help reduce the parent node's energy cost, as energy consumed for sensing and reporting sensory data usually is higher than that for relaying data for the child node. This way, the parent node could stay alive for a longer period of time to prevent the network from an early disconnection. Second, assigning more sensing duties to the shortest-lifetime child node can reduce the overall idling time of non-critical nodes. Thus, the non-critical nodes are able to use their energy more efficiently which extends (indirectly) the network lifetime. Figure 3 shows an example of how a parent node adjusts the sensing duties of itself and its child nodes.

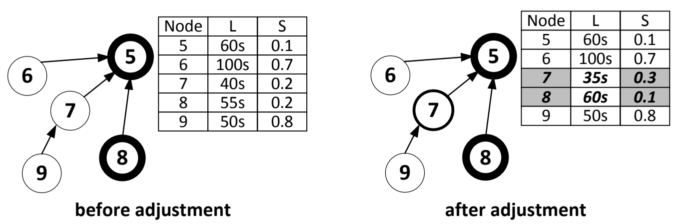


Fig. 3. An example of sensing duty adjustment among parent and child nodes. $\alpha = 2$, nodes 6, 7 and 9 are non-critical and nodes 5 and 8 are critical. Sensing duties are shifted from node 8 (critical node with the shortest lifetime) to node 7 (non-critical node with the shortest lifetime). After this change, node 7 may shift its sensing duties further to its own child node 9.

After the adjustment, the parent node records the updated sensing duties, and informs each child of the update by embedding it in the ACK during the next communication between the parent node and the child node.

D. Child Node Behavior

To facilitate the parent node's sensing duty adjustment, each child node i needs to compute and embed the value of L_i in its data packet. In particular, L_i is computed as:

$$L_i = \frac{e_i}{c_i}, \tag{2}$$

where e_i is node i's current residual energy and c_i is the node's current energy consumption rate estimated based on its sensing activities:

$$c_i = S_i \beta \theta_s + \sum_{j \in \mathcal{P}_i} f_{i \to j} \theta_{tx} + \sum_{k \in \mathcal{C}_i} f_{k \to i} \theta_{rx} + \theta_\epsilon. \tag{3}$$

After receiving an ACK from the parent, a child node adjusts its sensing duty according to the information embedded in the ACK. In case its sensing duty cannot be changed any further, the child node will try to accommodate the difference between its current sensing duty and the assigned new sensing duty during its next round of adjustment among itself and its own child nodes.

V. Performance Evaluation

We have evaluated the performance of DSS in terms of network lifetime using ns-2 simulations, and compared DSS with the following schemes:

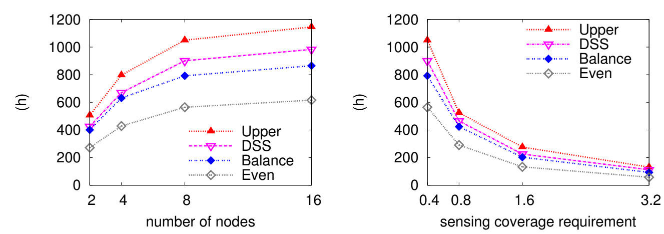
- Upper: a theoretical upper bound obtained from an NLP solver [25] of the formulation in Section III-C.
- Balance: a nodal lifetime balancing scheme, which assigns more sensing duties to nodes with a longer nodal lifetime, in order to maximize the minimal nodal lifetime in the network, regardless whether a node is a critical node or not.
- Even: a naive scheme to allocate an identical sensing duty to all nodes in the same monitoring area.

A. Simulation Setup

In the simulations, RI-MAC [22] is employed as the underlying MAC protocol, where the nodal wakeup interval is one second and the channel checking period is 7 ms, both according to the default setting of RI-MAC. The energy consumed for transmitting or receiving a packet is $\theta_{tx} = \theta_{rx} = 2\ mJ$ [23], while the energy consumed for sensing and generating a data packet is $\theta_s = 3\ mJ$. In the idle mode when a node is not sensing and the radio is turned off, the power consumption is $\theta_\epsilon = 80\ \mu W$ [18]. DSS is evaluated in networks with various topologies (line, star, or random), various number of nodes, and various levels of sensing coverage requirement. The initial nodal energy is uniformly distributed between 500 and 1500 Joules. Each point in the figures of simulation results is averaged over 100 runs.

B. Simulation Results

1) Line-topology networks: We first evaluate DSS in line-topology networks with a single monitoring area. We vary the number of nodes between 2 and 16, and the sensing coverage requirement between 0.4 and 3.2. Simulation results are plotted in Figure 4.



(a) $\alpha = 0.4$.

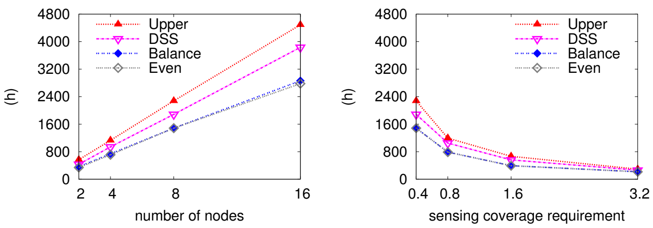
(b) The number of nodes is 8.

Fig. 4. Performance comparison in line-topology networks with a single monitoring area in the network.

We observe that the Even scheme yields the shortest network lifetime. This is due to the bottleneck effect in a line-topology network: nodes closer to the sink consume more energy as they have to relay data for other nodes in the network. With the Even scheme, all nodes are assigned with an identical sensing duty. Therefore, the bottleneck nodes run out of energy first, which disconnects the network early while other nodes may still have energy left.

In comparison, both DSS and Balance schemes perform well, and their performance is close to the derived theoretical upper bound. This is because they both attempt to shift the sensing duties away from the bottleneck nodes, hence allowing the bottleneck nodes to operate for a longer period of time. Furthermore, as DSS is aware of the nodes' criticality, it adopts a more intelligent sensing adjustment strategy than Balance to purposely extend the critical nodes' lifetime and consequently the network lifetime.

2) Star-topology networks: With a star topology, all nodes are one hop away from the sink. Hence, the Balance and Even schemes yield a similar performance as they allocate a similar level of sensing duty to each node. The results are shown in Figure 5.



(a) $\alpha = 0.4$. (b) The number of nodes is 8.

Fig. 5. Performance comparison in star-topology networks with a single monitoring area in the network.

Compared to the performances achieved in line-topology networks, DSS produces a significant improvement over the Balance strategy in star-topology networks. Moreover, as the number of nodes increases, the performance of DSS is improved further. For example, the improvement ratio is about 44% when there are 16 nodes in network. This is because DSS treats critical nodes and non-critical nodes differently. For non-critical nodes, DSS aims to improve their energy efficiency by scheduling them to perform sensing duties in a sequential manner. This way, the overall idling time of non-critical nodes is reduced, and the network lifetime is improved.

In comparison, the Balance strategy always tries to maintain a similar lifetime for all nodes, which may cause the network to operate with a lower energy efficiency and thus a shorter lifetime.

3) Random-topology networks: We have also evaluated DSS in random-topology networks, where 100 nodes are deployed in a 500 $\times$ 500 m field uniformly at random. The sink node is located at the center of the field. The maximal communication range of each node is 100 meters. The field is divided into grids, and each grid corresponds to a monitoring area. Figure 6(a) shows an example of the random-topology network with 9 monitoring areas.

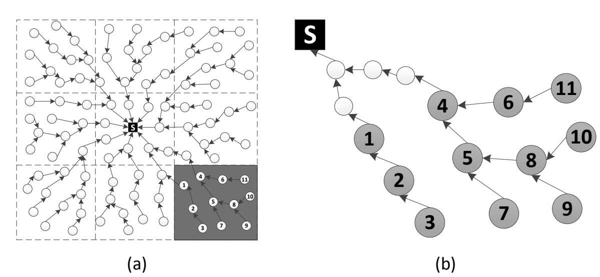


Fig. 6. (a) An example random-topology network with 9 monitoring areas and a sink node at the center. A routing tree rooted at the sink is shown in the figure. (b) An enlarged view of the bottom right monitoring area; the behaviors of nodes in this area are used for a trace study in Fig. 8.

Simulation results for random-topology networks are shown in Fig. 7. Not surprisingly, DSS outperforms both Balance and Even strategies, thanks to its dynamic adjustment of sensing duties according to nodes' criticality as well as nodal lifetime. Specifically, the improvement ratios of DSS over Balance and Even strategies are about 15-30% and 70-100%, respectively.

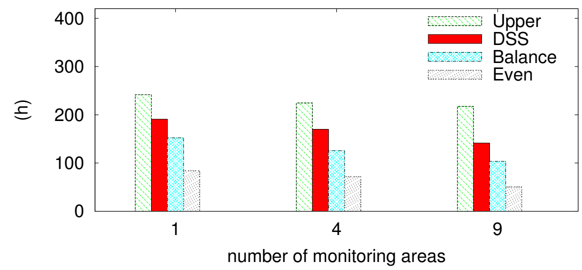


Fig. 7. Performance comparison in random-topology networks with multiple monitoring areas in the network. The overall sensing coverage requirement is 9; therefore, if the field is divided into x monitoring grids, the sensing coverage requirement in each area is $9/x$.

One interesting observation from Fig. 7 is that, as the field is divided into a larger number of monitoring areas, the performance of all evaluated schemes, including DSS, decreases. This phenomenon can be explained as follows. When there are more monitoring areas in the network, neighboring nodes may be deployed to monitor different areas, hence their sensing duties cannot be adjusted jointly; otherwise, the sensing coverage requirement in some areas may be violated. Therefore, less number of nodes may participate in the DSS process, resulting in a decreased lifetime performance. This observation also motivates the possible improvement of DSS to schedule sensing duties of nodes across monitoring areas, jointly with routing activity adjustment, which is part of our future work.

4) Trace study: Fig. 8 presents a detailed trace study on a particular simulation run for the bottom right monitoring area of the network shown in Fig. 6(a). It has 11 sensor nodes, and the sensing coverage requirement for the area is $\alpha = 1$. An enlarged view of the area is given in Fig. 6(b).

As shown in Fig. 8, at the beginning of the simulation, all nodes are assigned with an identical sensing duty. After

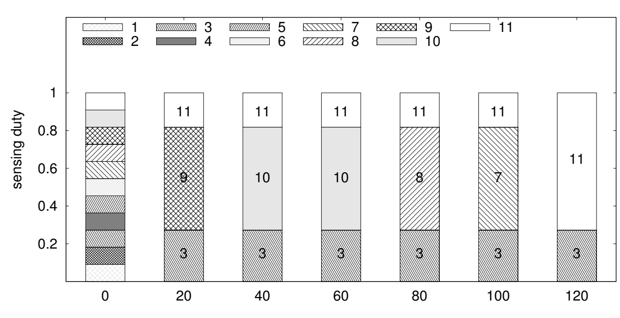


Fig. 8. Snapshots of sensing duty assignment to nodes 1 to 11 in the monitoring area shown in Figure 6(b), during a particular simulation run. X-axis represents the time instances when the snapshot is taken.

20 hours of running, three non-critical leaf nodes (3, 9, and 11) are assigned with much higher sensing duty than any other node in the same branch. After 40 hours, node 9 runs out of energy, and DSS reassigns its sensing duty to another non-critical leaf node 10 to maintain the required sensing coverage requirement. After 80 hours, node 10 also runs out of energy, and node 8 becomes a new leaf node. DSS continues to assign node 8 and then node 7 to sense in a sequential manner to reduce the idling time of non-critical nodes and improve the energy efficiency. After 120 hours, node 5 runs out of energy, and nodes 3 and 11 become the only two leaf nodes in the area. Note from Fig. 6 that the two branches (that nodes 3 and 11 belong to) report to the sink via different routes through different areas. Thus, the sensing duties could not be shifted between them. As a result, both nodes 3 and 11 continue to monitor the area, until both branches are disconnected from the routing tree.

VI. Conclusion

In this paper, we propose a distributed and lightweight sensing scheduling scheme, called DSS, to prolong the lifetime of a sensor network under practical requirements of sensing coverage and network connectivity. Different from lifetime-balancing schemes, DSS is unique in that it schedules less sensing duties to nodes that are critical for sensing coverage, but more to non-critical nodes even at the cost of losing them more quickly. As the sensing coverage and network connectivity requirements can be satisfied for a longer period of time, the network lifetime can be prolonged. The effectiveness and advantages of DSS are demonstrated using ns-2 simulations.

DSS was designed to work with a static data collection tree. In the future, we plan to improve the DSS design by taking into account the routing behaviors that also affect the network traffic distribution and hence the network lifetime, to overcome the limitations of DSS revealed in the experiments. Via cross-layer optimization, a joint routing and sensing design may prolong the network lifetime even further. In addition, how to incorporate more application specified constraints such as end-to-end data delivery delay into the design is another direction of future work.

References

[1] G. Anastasi, M. Conti, and M. Di Francesco, “Extending the Lifetime of Wireless Sensor Networks Through Adaptive Sleep,” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 351–365, August 2009.

[2] T. Arici and Y. Altunbasak, “Adaptive Sensing for Environment Monitoring Using Wireless Sensor Networks,” in *IEEE WCNC*, 2004.

[3] R. Braynard, A. Silberstein, and C. S. Ellis, “Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol,” in *EWSN*, 2006.

[4] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks,” in *IEEE INFOCOM*, 2006.

[5] M. Conti, G. Maselli, G. Turi, and S. Giordano, “Cross-Layering in Mobile Ad Hoc Network Design,” *IEEE Computer*, vol. 37, pp. 48–51, 2004.

[6] S. Cui, R. Madan, A. Goldsmith, and S. Lall, “Joint Routing, MAC, and Link Layer Optimization in Sensor Networks with Energy Constraints,” in *IEEE ICC*, 2005.

[7] I. Dietrich and F. Dressler, “On the Lifetime of Wireless Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 5:1–5:39, Feb 2009.

[8] E. J. Duarte-Melo and M. Liu, “Analysis of Energy Consumption and Lifetime of Heterogeneous Wireless Sensor Networks,” in *IEEE Globalcom*, 2002.

[9] A. Giridhar and P. R. Kumar, “Maximizing the Functional Lifetime of Sensor Networks,” in *ACM/IEEE IPSN*, 2005.

[10] Y. Gu and T. He, “Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links,” in *ACM SenSys*, 2007, pp. 321–334.

[11] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, “Lifetime and Coverage Guarantees Through Distributed Coordinate-free Sensor Activation,” in *ACM MobiCom*, 2009.

[12] V. Kawadia and P. R. Kumar, “A Cautionary Perspective on Cross Layer Design,” *IEEE Wireless Communications*, vol. 12, pp. 3–11, 2005.

[13] Q. Li, J. Aslam, and D. Rus, “Online Power-aware Routing in Wireless Ad-hoc Networks,” in *ACM MobiCom*, 2001.

[14] Z. Li, M. Li, and Y. Liu, “Towards Energy-Fairness in Asynchronous Duty-Cycling Sensor Networks,” in *IEEE INFOCOM*, 2012.

[15] Z. Li, Y. Peng, D. Qiao, and W. Zhang, “Joint Aggregation and MAC Design to Prolong Sensor Network Lifetime,” in *IEEE ICNP*, 2013.

[16] L. Lin, N. B. Shroff, and R. Srikant, “Asymptotically Optimal Power-Aware Routing for Multihop Wireless Networks with Renewable Energy Sources,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1021–1034, October 2007.

[17] A. Meier, M. Woehrle, M. Zimmerling, and L. Thiele, “ZeroCal: Automatic MAC Protocol Calibration,” in *IEEE DCOSS*, 2010.

[18] MEMSIC, “TelesB Mote Datasheet,” http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf.

[19] W. Mo, D. Qiao, and Z. Wang, “Mostly-Sleeping Wireless Sensor Networks: Connectivity, k-Coverage, and α-Lifetime,” in *The 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.

[20] J. Park and S. Sahni, “An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks,” *IEEE Transactions on Computers*, vol. 55, pp. 1048–1056, August 2006.

[21] Y. Peng, Z. Li, D. Qiao, and W. Zhang, “I2C: A Holistic Approach to Prolong the Sensor Network Lifetime,” in *IEEE INFOCOM*, 2013.

[22] Y. Sun, O. Gurewitz, and D. B. Johnson, “RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks,” in *ACM SenSys*, 2008.

[23] Texas Instruments, “CC2420: 2.4Ghz IEEE802.15.4/Zigbee-ready RF Transceiver,” http://focus.ti.com/lit/ds/symlink/cc2420.pdf, 2006.

[24] D. Tian and N. D. Georganas, “A Coverage-preserving Node Scheduling Scheme for Large Wireless Sensor Networks,” in *ACM WSNA*, 2002.

[25] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, pp. 25–57, March 2006.

[26] K. Wu, Y. Gao, F. Li, and Y. Xiao, “Lightweight Deployment-aware Scheduling for Wireless Sensor Networks,” *Mobile Networks and Applications*, vol. 10, no. 6, pp. 837–852, December 2005.

[27] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 36–72, August 2005.

[28] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiel, “pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols,” in *ACM/IEEE IPSN*, 2012.