Energy Minimization in Multi-Task Software-Defined Sensor Networks

Deze Zeng, *Member, IEEE*, Peng Li, *Member, IEEE*, Song Guo, *Senior Member, IEEE*, Toshiaki Miyazaki, *Senior Member, IEEE*, Jiankun Hu, *Member, IEEE*, and Yong Xiang, *Senior Member, IEEE*

Abstract—After a decade of extensive research on application-specific wireless sensor networks (WSNs), the recent development of information and communication technologies makes it practical to realize the software-defined sensor networks (SDSNs), which are able to adapt to various application requirements and to fully explore the resources of WSNs. A sensor node in SDSN is able to conduct multiple tasks with different sensing targets simultaneously. A given sensing task usually involves multiple sensors to achieve a certain quality-of-sensing, e.g., coverage ratio. It is significant to design an energy-efficient sensor scheduling and management strategy with guaranteed quality-of-sensing for all tasks. To this end, three issues are investigated in this paper: 1) the subset of sensor nodes that shall be activated, i.e., sensor activation, 2) the task that each sensor node shall be assigned, i.e., task mapping, and 3) the sampling rate on a sensor for a target, i.e., sensing scheduling. They are jointly considered and formulated as a mixed-integer with quadratic constraints programming (MIQP) problem, which is then reformulated into a mixed-integer linear programming (MILP) formulation with low computation complexity via linearization. To deal with dynamic events such as sensor node participation and vanishment, during SDSN operations, an efficient online algorithm using local optimization is developed. Simulation results show that our proposed online algorithm approaches the globally optimized network energy efficiency with much lower rescheduling time and control overhead.

Index Terms—Software-defined Sensor Network, Sensor Activation, Task Mapping, Sensing Rate Scheduling, Energy Efficiency

1 Introduction

Wireless Sensor Networks (WSNs) have been widely deployed for a wide span of applications including surveillance, tracking, and controlling. While many efforts have been made to enhance the applicability and performance of WSNs from different layers, they mainly consider the case that a WSN is dedicated for one sensing task. Such an application-specific WSN is prone to

1) high deployment cost: multiple WSNs for respective tasks may be deployed in the same area,
2) low service reutilization: different vendors develop their WSNs in an isolated manner without sharing common functionalities, and
3) difficult hardware recycling: altering existing code on single-task sensor nodes is difficult, highly error-prone, and costly [1].

Software-Defined Sensor Networks (SDSNs) emerge as a compelling solution to the above issues [1]. An SDSN consists of a number of sensor nodes whose functionalities can be dynamically configured by injecting different programs. A software-defined sensor node equipped by several different types of sensors is able to conduct different sensing tasks according to the programs deployed and activated. Such kind of sensor node prototypes have been practically realized. Miyazaki et al. [2], [3] implemented the software-defined sensor nodes that can dynamically change their functions upon specific sensing task requirements at runtime.

In this paper, we consider an SDSN as shown in Fig. 1, which consists of a sensor control server and a set of software-defined sensor nodes. Any node is integrated with multiple sensors of various types, e.g., ultrasonic sensor, photoelectric sensor, infrared sensor and etc., each of which is responsible for a specific sensing task for a corresponding group of targets in its sensing area. The targets are classified by the sensing tasks as depicted by different notations, i.e., ★, ▲, and ■ in Fig. 1. At each time instance, a sensor node can exactly sense one target. In software-defined sensor node, operation system is required for the management of the sensor resources [2]. Modern sensor node OS like TinyOS [4] usually supports multi-task that can execute independently and non-preemptively. Multiple tasks with different sensing targets may be issued to the SDSN. For each task, there is a related program. Only the sensors that have been loaded and activated with the corresponding program are able to sense the related targets within its coverage. For example, a sensor node can conduct vibration and

This work is supported by Strategic Information and Communications R&D Promotion Programme (SCOPE No. 121802001) and Grants-in-Aid for Scientific Research (No. 23500095).

- *D. Zeng is with School of Computer Science, China University of Geosciences, Wuhan, China.*
  E-mail: zengdeze@gmail.com
- *P. Li, S. Guo and T. Miyazaki are with the Department of Computer Science and Engineering, The University of Aizu, Japan.*
  E-mail: {pengli, sguo, miyazaki}@u-aizu.ac.jp
- *J. Hu is with the School of Engineering and Information Technology, The University of New South Wales, Canberra ACT 2610 Australia.*
  E-mail: J.Hu@adfa.edu.au
- *Y. Xiang is with the School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia.*
  E-mail: yong.xiang@deakin.edu.au

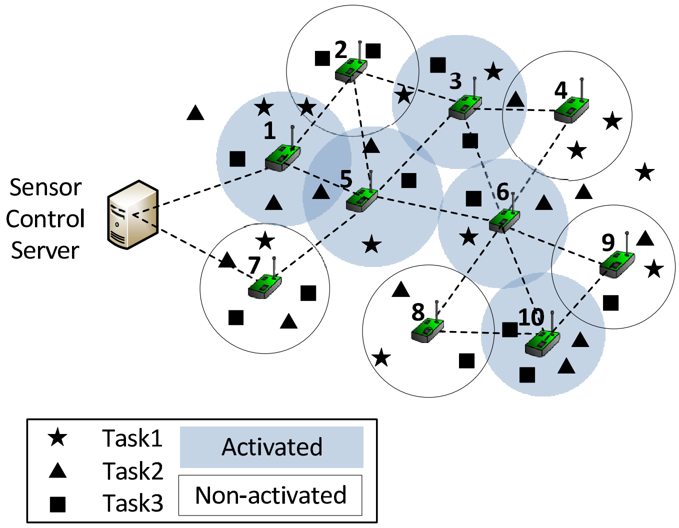


Fig. 1. An example of software-defined sensor network with three types of sensing targets

heat detection using photoelectric and infrared sensors, respectively, provided that both corresponding programs are loaded.

Energy efficiency is always a critical issue in WSNs with battery-powered nodes and SDSNs are without exception. Intuitively, the less sensors are activated, the less energy shall be consumed. However, a sensing task requires certain level of quality-of-sensing, e.g., coverage ratio, which is a commonly adopted quality-of-sensing metric describing the portion of targets covered by the reprogrammed sensors [5]. Ideally, for multi-task SDSNs, the computation and storage capacity limitations of sensors and the requirements of sensing tasks shall be jointly taken into account. Therefore, in this paper, we are motivated to study a minimum-energy sensor activation problem in multi-task SDSNs with guaranteed quality-of-sensing. We specially make the following contributions:

- To our best knowledge, we are the first to study the minimum-energy sensor activation problem with the consideration of schedulability and guaranteed quality-of-sensing in multi-task SDSNs. We first derive the effective sensing rate that can be achieved by collaborative sensing in closed form, based on which we formulate the minimum-energy sensor activation problem as a mixed-integer with quadratic constraints programming (MIQP) problem by jointly considering sensor activation and task mapping.
- Based the formulation, we further propose an efficient online algorithm to deal with dynamic events during runtime of SDSNs.
- Through extensive simulation based studies, we show the high efficiency of the proposed algorithm by the fact that it achieves almost the same power efficiency as global optimization but with much lower rescheduling time and control overhead.

The rest of this paper is organized as follows. Section 2 provides a brief overview of related work. Section 3 states our network model. The MIQP formulation and its linearization to MILP are proposed in Sections 4 and 5, respectively. Section 6 gives an MILP-based online algorithm to deal with dynamic events in SDSNs. Section 7 shows our performance evaluation results. Finally, Section 8 concludes our work.

2 Related Work

2.1 Software-Defined Sensor Networks

It is widely agreed that SDSN is an inevitable developing trend of WSN with the fast development of various information and communication technologies [1]. A few pioneering investigations have been made in the literature.

Rossi et al. [6] present their design and implementation of SYNAPSE++, a fountain codes based code dissemination strategy in WSNs. Miyazaki et al. [2], [3] recently realize Die-hard Sensor Network (DSN), which consists of sensor nodes that can dynamically adjust their sensing functions by turning on the corresponding sensing unit. Shafi et al. [7] propose an efficient over-the-air-programming technique called Queen's Differential that can detect and explore the similarities between the new problem and the older one to reduce the program size to be written.

Existing work has successfully proved the feasibility of SDSN. In practice, it is essential to achieve high energy efficiency without sacrificing quality-of-sensing. Our work put forward the study on SDSN from the perspective of sensor resource allocation in SDSNs, with a special emphasis on the multi-task scheduling.

2.2 Coverage Problem in WSNs

For WSNs, an important issue is on the quality-of-sensing, e.g., coverage, which has been widely addressed in the literature from different aspects, e.g., placement, sensor scheduling, etc.

Slijepcevic et al. [8] present a SET-K cover problem in which sensors are organized into mutually exclusive sets and each set is equivalent to a subarea monitored by a corresponding sensor. Chakrabarty et al. [9] consider a problem on how to place sensor nodes in a grid to minimize the total cost and formulate it into an integer linear programming (ILP) problem, which is then solved by a divide-and-conquer method. They assume that both sensor nodes and sensing targets are on a grid. Huang et al. [10] discuss the way to determine how well a sensor network is monitored or tracked by sensors by formulating two decision problems for unit-disk coverage and non-unit disk coverage, respectively. Cheng et al. [11] address the minimum coverage beach problem NP-Complete under bandwidth constraints. It is then formulated into an ILP problem and solved using heuristic algorithms.

One branch on studying the coverage problem in WSNs is on how to activate the sensor nodes to achieve certain level of coverage towards different objectives. Kar et al. [12] address the problem of how rechargeable sensor nodes should be activated dynamically so as to maximize a generalized system performance. Recently, Kasbekar et al. [13] design a polynomial-time distributed algorithm for maximizing the lifetime of the network by designing a schedule on the sequence of sensors that shall be activated in every time slot. Hsin et al. [14] investigate the network coverage on a point-of-interest (PoI) in low duty-cycled collaborative sensors and present the role-alternating, coverage-preserving (RACP) algorithm to minimize the probability that any given point is not covered by an active sensor. Recently, the stochastic event detection in low duty-cycle collaborative sensors, with and without the consideration of network connectivity has been studied in [15] and [16], respectively.

The key difference of SDSNs to conventional single-task WSNs is on the reprogrammability of sensors. The coverage issue is also highly related to the reprogramming energy consumption. Our work takes the coverage requirement as an input to the minimum-energy reprogramming problem. In addition, we fill in the blank in the field of collaborative sensing by presenting a new analytical framework on the "effective sensing rate".

2.3 Task Scheduling in WSNs

It is regarded that task-based systems are needed to provide services to entities outside the next-generation WSNs. Task scheduling in WSNs therefore attracts many interests in the literature.

Tian et al. [17] discuss a task mapping and scheduling solution for energy-constrained applications in WSNs, energy-constrained task mapping and scheduling (EcoMapS), which incorporates channel modeling, concurrent task mapping, communication and computation scheduling, and sensor failure handling. Voinescu et al. [18] consider the allocation of tasks onto different wireless nodes with the consideration energy constraints, compatibility of tasks to a given node and network topology. Habib et al. [19] model the sensing, processing and transmitting tasks as a directed acyclic graph (DAG) per sensor and group DAGs into a super task-flow-graph (STFG), within which data aggregation problem is solved by scheduling the tasks. Later on, Edalat [20] apply game theory and propose an auction-based distributed task allocation in resource-constrained WSNs to maximize the overall network lifetime. Recently, Li et al. [21] consider collaborative processing in WSNs and study how to schedule tasks in a systematic way, including assigning tasks to sensor nodes, and determining their execution and communication sequence, to minimize the overall energy consumption and balance the workloads. Later, they [22] present the Resource Allocation in Heterogeneous WSNs (SACHSEN) algorithm to effectively and dynamically coordinate the sharing of the available resources to optimize resource utilization while meeting application requirements. Farias et al. [23] propose a task scheduling algorithm that explores the characteristics of applications with common tasks in shared sensor and actuator networks to improve the system energy efficiency in order to prolong the network lifetime.

TABLE 1
Notations

S	the sensor node set
s	sensor node $s \in S$
(x_s, y_s)	position of the sensor node s
a_s	whether sensor node s is activated or not
T	sensing task set
t	sensing task $t \in T$
G_t	the set of targets of task $t \in T$
f_t	the minimum sampling rate requirement for each target of task $t \in T$
c_t	the required operation duration for each target of task $t \in T$
s_t	program size of task $t \in T$
g	a sensing target $g \in G_t$ of task $t \in T$
r_t	the sensing range of task $t \in T$
α_{st}	whether sensor node s is scheduled for sensing task $t \in T$
β_{stg}	whether sensor node s can sense target g of sensing task $t \in T$
γ_{tg}	whether target g of task t is covered by at least one sensor
υ_{stg}	whether target g of task t is within the sensing range of sensor node s
δ_t	coverage ratio requirement of sensing task $t \in T$
P_a	power consumption of an activated sensor
P_s	power consumption of a sleeping sensor

Regarding the sensing quality, Xu et al. [24] study multiple-application allocation in shared sensor networks to maximize the quality-of-monitoring (QoM) subject to resource constraints (e.g., memory and bandwidth). Kapoor et al. [25] study task scheduling and sensor node resource allocation in a shared WSN hosting multiple applications, each of which varies in the requirements on the number of sensors required. Tang et al. [26] define a non-decreasing submodular function to measure the QoM by considering the correlation in sensed data and propose distributed scheduling schemes to pursue high QoM in a fixed duty-cycling shared sensor network.

However, none of existing studies discusses the task mapping problem with the consideration of schedulability. To our best knowledge, we are the first to take this issue into WSN task scheduling.

3 System Model

We consider an SDSN like the one shown in Fig. 1 where each sensor node is equipped with multiple sensors with different sensing capabilities, e.g., temperature, humidity, light, vibration, etc. Such an SDSN can run multiple tasks, each of which is explicitly specified with a set of sensing targets and the quality-of-sensing requirement, e.g., minimum *coverage ratio*. In this paper, we are mainly interested in point coverage, where a target is a detectable spatial point-of-interest (PoI) with location information [5], [27], [28]. For example, in a structural health monitoring (SHM) task, a target refers to a vulnerable point that may be critical to the health of a monitored structure, e.g., a skyscraper or a bridge, as discussed in [29]. Let S and T denote the sensor set and task set, respectively. For a task $t \in T$, its sensing target set is represented by P_t and the required coverage ratio is denoted by δ_t .

Without loss of generality, we assume that both sensor nodes and the sensing targets are randomly distributed in the network area. Target $g \in G_t$ of a task $t \in T$ can be monitored by sensor $s \in S$ only if residing within the sensing range of s. This fact can be expressed by notations:

$$\upsilon_{stg} = \begin{cases} 1, \text{ if target } g \text{ of task } t \text{ is within the sensing} \\ \quad \text{range of sensor node } s \ , \\ 0, \text{ otherwise.} \end{cases}$$

Note that, a sensor node may have different sensing ranges for different tasks. Once the locations of sensor nodes and the sensing targets are known, the values of υ_{stg} will be determined.

In SDSNs, only the sensor nodes $s \in S$ loaded with a program for task $t \in T$ can sense t's targets. Therefore, to conduct a sensing task on a sensor node first requires that the corresponding program is stored on the sensor node. Different programs are with different program sizes and hence are with diverse storage requirements. We consider a homogeneous sensor network that all the sensors are with the same storage capacity, which is normalized as 1.0 throughout this paper. Consequently, let $z_t, 0 < z_j \leq 1, t \in T$ denote the normalized program size of task t.

To ensure the sensing accuracy to a target, we must guarantee the minimum sensing rate requirement, which refers to how often a sensing operation shall be conducted to a target. The sensing rate is usually specified in the task requirement. Only the minimum sensing rate to a target is satisfied, we can say that the target is covered. Besides, each sensing operation also needs certain sensing and computation duration. The need for non-negligible sensing durations to obtain useful information is due to noises in the measurement process and the probabilistic nature of the phenomena under observation [30]. For a task $t \in T$, let us use f_t and c_t to denote the required minimum sensing rate and duration to its target, respectively. For example, to sense the vibration at a PoI (i.e., target) of a SHM task t, we need sensing rate and duration as 300Hz and 1ms, respectively, i.e., $f_t = 300$Hz and $c_t = 1$ms. We treat a target g of sensing task t as "covered" provided that "the effective sensing rate", denoted as f^e_{tg}, from multiple sensors exceeds the minimum sensing rate requirement f_t. Any involved sensor s independently makes the corresponding sensing activity following a Poision process with a rate f_{stg}.

Multiple tasks need to be handled by an SDSN at the same time. A subset of the sensor nodes in the network shall be activated for these tasks. It has been widely proved that careful scheduling on the sensor activation is a promising way to conserve the energy consumption [31]–[34]. Therefore, a sensor could be in activated mode or sleeping mode, with power consumption P_a and P_s, respectively. In an SDSN, a sensor node may be loaded with multiple programs for different tasks. Therefore, we are interested in which sensors shall be activated and which tasks shall be assigned to each of them to minimize the sensing power consumption while guaranteeing the quality-of-sensing for each task.

For the ease of reading, Table 1 summarizes all the notations to be used throughout this paper.

4 Problem Formulation

4.1 Coverage Ratio Constraints

We first consider the quality-of-sensing requirement for a sensing task in SDSNs. A sensor $s \in S$ can sense the target $g \in G_t$ of task $t \in T$ if and only if 1) target t is within the sensing range of s, i.e., $\upsilon_{stg} = 1$, and 2) sensing task t is scheduled on s. We define a binary variable α_{ij} to denote whether sensor s is scheduled with task t or not as follows:

$$\alpha_{st} = \begin{cases} 1, \text{ if sensor } s \text{ is scheduled with task } t, \\ 0, \text{ otherwise.} \end{cases}$$

By letting β_{stg} denote whether sensor s is able to sense target g of task t, we have

$$\beta_{stg} = \alpha_{st}\upsilon_{stg}, \forall s \in S, t \in T, g \in G_t. \tag{1}$$

For any sensing target counted as covered (i.e., $\gamma_g = 1, \forall t \in T, g \in G_t$), the minimum sensing rate to it must be reserved to ensure the sensing accuracy. The corresponding program shall be invoked periodically according to the task sensing rate requirement. Note that the sensing rate f_{stg} for a target is first determined by β_{stg} as

$$0 \leq f_{stg} \leq \beta_{stg} \cdot f_t, \forall s \in S, t \in T, g \in G_t, \tag{2}$$

which indicates that only when sensor s is able to sense target g of task t, f_{stg} can be allocated with a value larger than 0.

Multiple sensors may be able to sense one target cooperatively. For example, as shown in Fig. 1, the target of *Task 2* in black triangle can be sensed by both sensor 1 and sensor 5 in their overlapped coverage. Note that the effective sensing rate is not simply obtained by summing up the sensing rates contributed by the collaborative sensors, because any sensing event that meets an on-going one for the same target will be considered duplicate and should be ignored. The derivation of the effective sensing rate is given in the theorem below.

Theorem 1: The effective sensing rate in collaborative sensing for target g of sensing task t is

$$f^e_{tg} = \frac{\sum_{s \in S} f_{stg}}{1 + c_t \sum_{s \in S} f_{stg}}. \tag{3}$$

Proof: Recall that the sampling events are triggered as a Poisson process by each sensor independently. The overall sensing activities can thus be considered as a compound Poisson process with rate $f = \sum_{s\in S} f_{stg}$. When a sensing activity of task t (with a fixed duration c_t) is on-going, any newly triggered event in this duration will be considered as duplicated. Therefore, the collaborative sensing for target g of task t can be modeled as an M/D/1/1 queue. According to the transient analysis by Garcia et al. on M/D/1/N queue [35], we can derive the stationary probability distribution function for our M/D/1/1 queue as

$$\begin{aligned} \pi_0 &= \frac{1}{1+\rho}, \\ \pi_1 &= 1 - \frac{1}{1+\rho}, \end{aligned} \tag{4}$$

where

$$\rho = c_t \sum_{s\in S} f_{stg}. \tag{5}$$

A sensing event is considered as effective only when the "queuing system" is in state π_0. Therefore, the effective arrival rate f_{tg}^e can be calculated as

$$f_{tg}^e = f\pi_0 = \frac{\sum_{s\in S} f_{stg}}{1 + c_t \sum_{s\in S} f_{stg}}. \tag{6}$$

□

Based on the result from Theorem 1, we define a binary variable γ_{tg} to represent if the effective sensing rate f_{tg}^e exceeds the required sensing rate f_t, i.e.,

$$\gamma_{tg} = \begin{cases} 1, \text{ if } \dfrac{\sum_{s\in S} f_{stg}}{1 + c_t \sum_{s\in S} f_{stg}} \geq f_t, & \forall t \in T, g \in G_t. \\ 0, \text{ otherwise.} \end{cases} \tag{7}$$

We can rewrite (7) by the following two inequalities:

$$\gamma_{tg} \cdot (\sum_{s\in S} f_{stg} - f_t - c_t f_t \sum_{s\in S} f_{stg}) \geq 0, \forall t \in T, g \in G_t, \tag{8}$$

and

$$(1-\gamma_{tg}) \cdot (f_t + c_t f_t \sum_{s\in S} f_{stg} - \sum_{s\in S} f_{stg}) \geq 0, \forall t \in T, g \in G_t \tag{9}$$

It can be verified that if $\sum_{s\in S} f_{stg} - f_t - c_t f_t \sum_{s\in S} f_{stg} \geq 0$, i.e., $\frac{\sum_{s\in S} f_{stg}}{1+c_t\sum_{s\in S} f_{stg}} \geq f_t$, then γ_{tg} must be 1 in order to satisfy (9). Similarly, if $\sum_{s\in S} f_{stg} - f_t - c_t f_t \sum_{s\in S} f_{stg} \leq 0$, i.e., $\frac{\sum_{s\in S} f_{stg}}{1+c_t\sum_{s\in S} f_{stg}} \leq f_t$, then γ_{tg} must be 0 in order to satisfy (8).

To ensure the quality-of-sensing for each sensing task, e.g., $t \in T$, a coverage ratio δ_t shall be achieved. This leads to the following constraints:

$$\sum_{g\in G_t} \gamma_{tg}/|G_t| \geq \delta_t, \forall t \in T. \tag{10}$$

4.2 Schedulability Constraints

A sensor node may have multiple targets to sense. A sensing target $g \in G_t, \forall t \in T$ requires certain sensing rate f_{stg} and duration c_t on a sensor node $s \in S$.

According to [36], the following constraints must be satisfied for all the sensor nodes:

$$\sum_{t\in T} \sum_{g\in G_t} c_t \cdot f_{stg} \leq 1, \forall s \in S. \tag{11}$$

to ensure the multi-task schedulability.

4.3 Storage Constraints

Due to the sensor node storage capacity limitation, the total storage requirement for all the tasks mapped onto a sensor node shall not exceed its storage capacity. Then, we have:

$$\sum_{t\in T} \alpha_{st} \cdot z_t \leq 1, \forall s \in S \tag{12}$$

4.4 Problem Formulation

If a sensor node is scheduled for at least one task, it must be activated to conduct the sensing operations. Let binary $a_s, s \in S$ denote whether a sensor node is activated or not, i.e.,

$$a_s = \begin{cases} 1, \text{ if sensor } s \text{ is activated,} \\ 0, \text{ otherwise.} \end{cases}$$

Then, we have:

$$\sum_{t\in T} \alpha_{st}/|T| \leq a_t \leq \sum_{t\in T} \alpha_{st}, \forall s \in S. \tag{13}$$

Obviously, from (13), we can see that $a_s \equiv 1$ if $\exists t \in T, \alpha_{st} = 1$ and $a_s \equiv 0$ only if $\forall t \in T, \alpha_{st} = 0$.

Our objective to minimize the total sensing power consumption is equivalent to minimizing the number of sensors that shall be activated, i.e., $\sum_{i\in I} a_i$. By summarizing all the above issues together, we may obtain an **MIQP** formulation as: Our objective of minimizing the total sensing power consumption can be represented as $\sum_{s\in S} a_s P_a + (1-a_s)P_s$. By summarizing all the above constraints together, we obtain an **MIQP** formulation as:

$$\begin{aligned} &\text{MIQP:} \\ &\quad \min: \sum_{s\in S} a_s P_a + (1-a_s)P_s, \\ &\quad \text{s.t.}: (1), (2), (8) - (13). \end{aligned}$$

5 Linearization

It has been proved that MIQP is NP-hard to solve [37]. Fortunately, we observe that the constraints (8) and (9) are nonlinear because of the products of variables. To linearize these constraints and lower the computation complexity, we define a new variable u_{stg} as follows:

$$u_{stg} = f_{stg}\gamma_{tg}, \forall s \in S, t \in T, g \in G_t, \tag{14}$$

which can be equivalently replaced by the following linear constraints:

$$0 \leq u_{stg} \leq f_{stg}, \forall s \in S, t \in T, g \in G_t, \quad (15)$$

$$f_{stg} + \gamma_{tg} - 1 \leq u_{stg} \leq \gamma_{tg} f_j, \forall s \in S, t \in T, g \in G_t. \quad (16)$$

The constraints (8) and (9) then can be written in a linear form as

$$\sum_{s \in S} u_{stg} - \gamma_{tg} f_t - c_t f_t \sum_{s \in S} u_{stg} \geq 0, \forall t \in T, g \in G_t, \quad (17)$$

and

$$\begin{aligned} & f_t + c_t f_t \sum_{s \in S} f_{stg} - \sum_{s \in S} f_{stg} \\ & - \gamma_{tg} f_t - c_t f_t \sum_{s \in S} u_{stg} + \sum_{s \in S} u_{stg} \geq 0, \forall t \in T, g \in G_t, \end{aligned} \quad (18)$$

respectively.

Now, we can linearize the **MIQP** problem into a mixed-integer linear programming (**MILP**) as

$$\begin{aligned} & \text{MILP:} \\ & \min: \sum_{s \in S} a_s P_a + (1 - a_s) P_s, \\ & \text{s.t.}: (1), (2), (10) - (13), (15) - (18). \end{aligned}$$

6 Online algorithm

In SDSNs, there are two kinds of network dynamics, referring to applications and sensor nodes, respectively. Applications may arrive during network operation and existing sensor nodes may vanish because of energy depletion or other events, and new nodes may be deployed. In this section, we present online algorithm to deal with these dynamic events.

6.1 Application Dynamics

As an SDSN may be released to tenants, who may submit new application or application requirement to sensor control server periodically. When sensor control server receives such an request, different from initial deployment, it shall consider the resource availability in the network as some nodes have already been deployed with certain applications. Fortunately, the sensor control server has global information of the whole network, with which it is able to apply global optimization similar to the initial deployment as discussed above.

When a new task t' comes, the control server updates the task set as $T = T \cup \{t'\}$ and make new task assignment decisions $a_{st}, \forall s \in S, t \in T$ for the new task set. The control server shall still consider coverage ratio constraints, schedulability constraints and storage constraints as discussed in Section 4. Besides, to avoid task migration on the sensor nodes that have been deployed with tasks, we further incorporate the following task-reserving constraints as

$$a_{st} \geq a_{st}^{\text{old}}, \forall s \in S, t \in T, \quad (19)$$

where a_{st}^{old} denotes the task assignment decision before the arrival of new application t'. 19 indicates that if task t has been assigned to node s, i.e., $a_{st}^{\text{old}} = 1$, such assignment shall still be reserved after resource allocation for the new task. Note that we do not enforce the reserving on the sensing frequency and therefore it is possible to tune the sensing frequencies of the assigned tasks on sensors so as to accommodate the new task, if necessary. Summing up all the constraints, we get the MILP for new application as

$$\begin{aligned} & \text{MILP-NEW-APP:} \\ & \min: \sum_{s \in S} a_s E_a + (1 - a_s) E_s, \\ & \text{s.t.}: (1), (2), (10) - (13), (15) - (19). \end{aligned}$$

6.2 Sensor Node Dynamics

On the other hand, in a practical wireless sensor network, sensor nodes are powered by batteries with limited capacity, and usually they cannot be recharged. Therefore, new nodes will be deployed to compensate the portion of sensors that have exhausted their batteries. In this section, we consider a dynamic network where existing nodes will vanish because of energy depletion, and new nodes will be deployed periodically. An intuitive method to deal with network dynamic is to apply the global optimization proposed in last section each time when joining or leaving events happen. Although such a method always leads to the minimum number of active sensors, it suffers from three weaknesses. First, frequent execution of global algorithm incurs high computational complexity at the sink. Second, the sink needs to collect the information of joining and leaving nodes for global optimization, and then deploys the results to all nodes, which would generate too many control messages that also consume a large amount of energy. Finally, the control messages between sink and sensors are exchanged over a multi-hop network, and the resulting delay cannot be ignored, especially in a large wireless sensor network where packets may travel through hundreds of hops from sink to the farthest node.

Instead of adopting global optimization to deal with network dynamic, we propose an online algorithm using local optimization with low complexity.

6.3 Participation

We first consider the case that a sensor node is deployed as a participator to the network. Although letting this node stay inactive will not degrade the quality-of-sensing, many opportunities of reducing the number of active nodes are missed. As an example shown in Fig. 2, two targets are sensed by two active sensor nodes A and B, respectively. When a new node C is deployed and its sensing range is large enough to cover two targets, we can deactivate nodes A and B to keep only one active sensor. To exploit such opportunities for performance improvement when new nodes are deployed, we

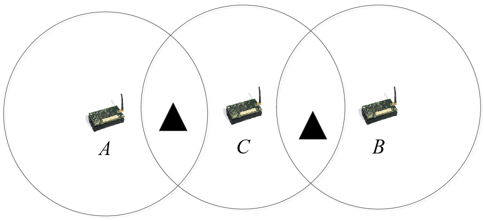


Fig. 2. An example of participation.

propose an online algorithm to deal with participation events.

Step 1: When a new node x is deployed, it first registers its information including the location, resource capacity, sensing units to the sensor control server.

Step 2: With the registered information, the sensor control server can discover the potential sensing targets within its sensing range. Centered at these potential targets, the sensor control server can also derive a set of sensor nodes that can reach these targets, e.g., sensor nodes A and B in Fig. 2.

Step 3: Based on the above information, the sensor control server can establish a subgraph including a set S' of sensors, and a set G'_t of targets for each task $t \in T$, as well as the corresponding quality-of-sensing δ'_t that should be satisfied in the subgraph, i.e.,

$$\delta'_t \equiv \sum_{g \in G'_t} \bar{\gamma}_{tg}/|G'_t|, \tag{20}$$

where $\bar{(\cdot)}$ denotes existing value of any variable $(\cdot)$.

Step 4: The sensor control server then solve a local optimization on the subgraph to reduce the number of active sensor nodes while guaranteeing the quality-of-sensing of each task. The formulation of local optimization **Local_OPT** is shown as follows.

$$\begin{aligned}
\min: & \sum_{s \in S'} a_s P_a + (1-a_s)P_s \\
\text{s.t.}: & 0 \le f_{stg} \le \alpha_{st} \cdot \upsilon_{stg} \cdot f_t, \forall s \in S', t \in T, g \in G'_t \\
& \sum_{g \in G'_t} \gamma_{tg}/|G'_t| \ge \delta'_t, \forall t \in T \\
& \sum_{t \in T} \sum_{g \in G'_j} c_t \cdot f_{stg} \le 1, \forall s \in S' \\
& \sum_{j \in J} \alpha_{ij} \cdot s_j \le 1, \forall i \in I' \\
& \sum_{j \in J} \alpha_{ij}/|J| \le a_i \le \sum_{j \in J} \alpha_{ij}, \forall i \in I' \\
& 0 \le u_{ijk} \le f_{ijk}, \forall i \in I', j \in J, k \in K'_j, \\
& f_{ijk} + \gamma_{jk} - 1 \le u_{ijk}, \forall i \in I', j \in J, k \in K'_j \\
& u_{ijk} \le \gamma_{jk} f_j, \forall i \in I', j \in J, k \in K'_j \\
& \sum_{i \in I'} u_{ijk} \ge \gamma_{jk} \cdot f_j, \forall j \in J, k \in K'_j \\
& (1-\gamma_{jk}) \cdot f_j \ge \sum_{i \in I'} (f_{ijk} - u_{ijk}), \forall j \in J, k \in K'_j
\end{aligned}$$

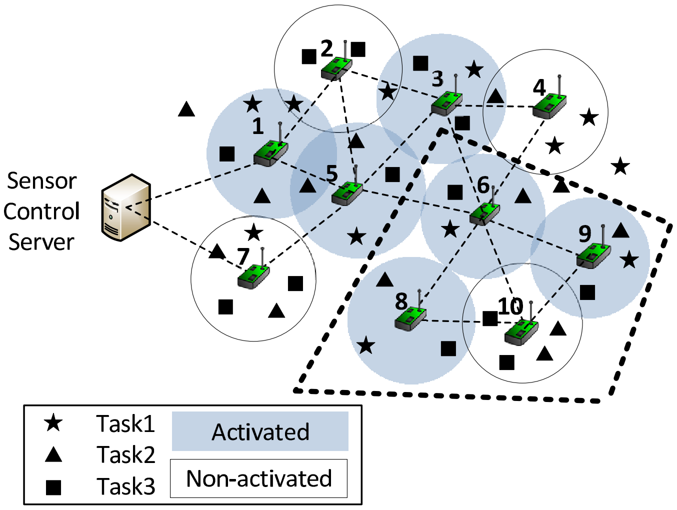


Fig. 3. New activation after the vanishment of node 10

Since the number of variables and constraints in local optimization is very limited, it can be easily solved even by resource-constrained sensor nodes.

Step 5: After solving the local optimization, if node x needs to become active such that other two or more sensors can be deactivated, it broadcasts the optimization results to its neighbors to adjust their sensing strategy. Otherwise, node x becomes inactive, and sensors in set I' still use the existing sensing strategy.

6.4 Vanishment

We then consider that case that a sensor node vanishes from the network because of energy depletion. We also adopt a local optimization to compensate the sensing rate of the vanished node with the minimize number of sensors. However, if the local sensor nodes are not able to guarantee the quality-of-sensing, we resort to the global optimization at the sink. The corresponding process are elaborated as follows.

Step 1: When the residual energy of a node x is under a threshold, it sends a message to inform the sink. The threshold is set when the sensor is deployed, and it should guarantee that the residual energy is enough the finish all operations in local optimization.

Step 2: Node x also adopt the local optimization Local_OPT on the subgraph to determine the sensing strategies of its neighbors after its vanishment. Note that node x is excluded from the set I' in the subgraph. For example, as shown in Fig. 3 where sensor node 10 is vanishing, four targets of *Task* 2 and Task 3 become uncovered after the vanishment of 10. A subgraph consisting of one-hop neighbors of node 10 is circled by the dashed quadrangle. Via applying Local_OPT to the subgraph, node 8 and 9 will be activated to ensure the coverage requirement.

Step 3: If the Local_OPT returns a feasible solution, node x broadcasts the optimization results to its neighbors, and then keeps working until its battery is exhausted. Otherwise, it sends a message to inform the sink about the failure of local optimization. Sink node will use a global optimization to guarantee the quality-of-sensing.

7 Performance Evaluation

In this section, we present our simulation-based performance evaluation results on the efficiency of our proposed online algorithm. We consider a 100×100 sensor network area randomly deployed with a number of software-defined sensor nodes. The default number of sensor nodes is set as 400. Unless otherwise stated, these tasks are developed with 100 targets for each randomly distributed within the network. The corresponding program sizes for the three tasks are normalized as 0.3, 0.4 and 0.4. Both sensing rate requirement and duration are uniformly distributed for all tasks, with default expectation 100Hz and 1ms, respectively. The power consumption of a sensor in activated mode and sleeping mode is set as $P_a = 12.0$mW and $P_s = 270\mu$W, respectively, which have been verified and adopted in [33]. The default sensing range is set as 6m. The online algorithm presented in Section 6 is implemented in our simulator, where node participation and departure events are randomly generated at runtime, and each triggers both "local" and "global" results obtained by our online algorithm global optimization, respectively. All the MILPs are solved using commercial solver Gurobi optimizer [38].

7.1 Effective Sensing Rate

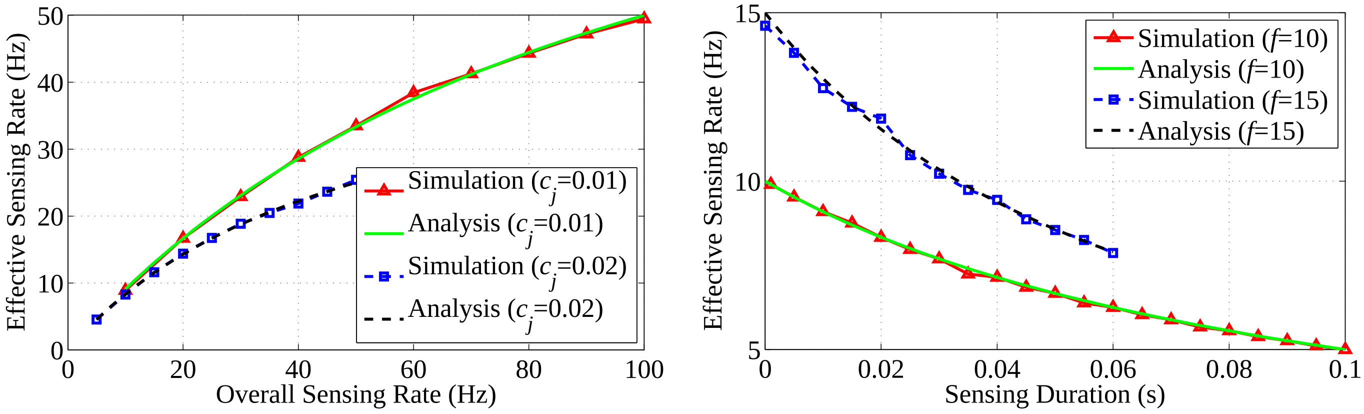
Before reporting our performance evaluation results, we first present the experimental results on the effective sensing rate that we can derive in Theorem 1. In the simulation, for a specific target of task t, the number of effective sensing events is counted under a certain overall sensing rate f. We first notice that the same result is achieved for the same overall sensing rate, no matter how it is distributed among the collaborative sensors. Such observation is consistent with the conclusion of compound Poisson process. Furthermore, Fig. 4(a) shows that the effective sensing rate increasing sublinearly with the overall sensing rate. This is because higher overall sensing rate indicates higher probability of duplicated sensing, under the same sensing duration. Fig. 4(b) shows the effective sensing rate as a non-linear decreasing function of the sensing duration. When $c_t = 0$, the effective sensing rate equals to the overall sensing rate as all the sensing operations are effective, while when $c_t f = 1$, the effective sensing ratio drops as low as 50%, e.g., around 5.0 when $f = 10$Hz. For the similar reason, longer sensing duration implies more duplicated sensing under the same sensing rate and hence lower effective sensing rate.

7.2 Rescheduling Time

To study the efficiency of our proposed online algorithm, we consider a SDSN randomly deployed with 400 sensor nodes. After the deployment, we simulate the sensor node dynamics as a Poisson process with average rate 0.1 events per time unit. The rescheduling time (i.e., the calculation time for the new solution) is investigated on a server configured with i7 quad-core 3.4GHz CPU, 8GB memory and Python 2.7.5. As a case study, Fig. 5(a) shows the instant rescheduling time for each dynamic event along 1000 time units. Obviously, in most cases, "local" requires much lower rescheduling time, compared to "global" one. Recall that our proposed online algorithm requires only local network information while the global one always ask for the information of the entire network. As a result, although both global algorithm and our local algorithm are with similar formulation, the smaller input (i.e., subgraph) makes our proposed online algorithm have much lower computation complexity than the global one requiring full graph information. However, in few cases, we also notice that "local" has longer rescheduling time than "global" one. For example, at time 152.82, the rescheduling time of "global" and "local" is 21.1s and 21.6s, respectively. This is because we cannot find a feasible solution via "local" optimization and have to resort to "global" one. Longer rescheduling time is thus incurred. Furthermore, one may also notice that the rescheduling time is not stable along the time line. The main reason is that the number of involving sensors and targets varies quite much from case to case in all randomly generated node participantion/departure events. To clearly compare the rescheduling time of our local optimization and the global one, we further plot the cumulative distribution function (CDF) of the rescheduling time in Fig. 5(b). Fig. 5(b) gives the CDF of the instant rescheduling time shown in Fig. 5(a). We can see that "local" exhibits outstanding rescheduling time advantage over the "global". For example, with probability around 60%, "local" can achieve a rescheduling time less than 0.323s while the one for "global" is 18.78s. According to the above performance evaluation results, we can conclude that our online algorithm indeed outperforms global optimization in rescheduling computation complexity.

7.3 Power Efficiency

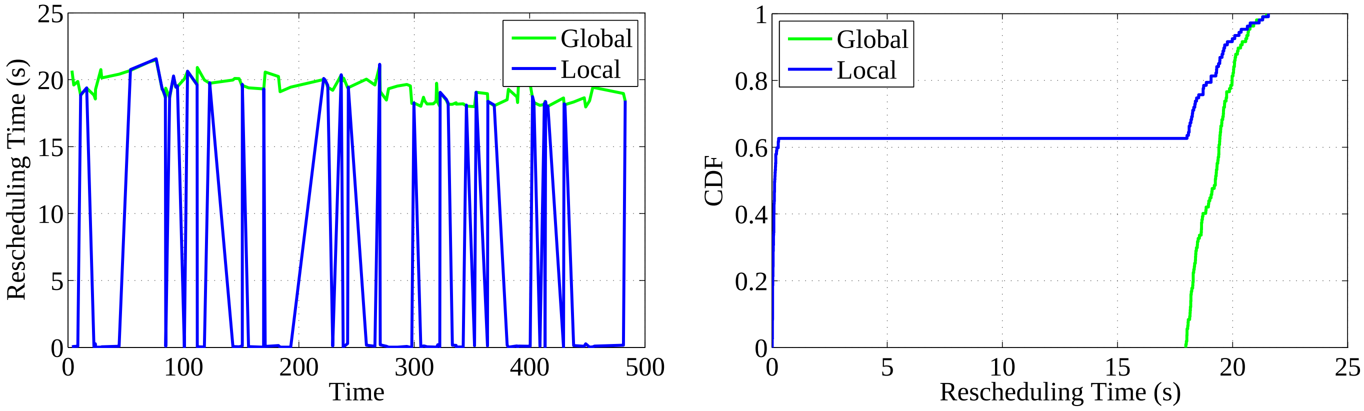
Although our online algorithm has advantages on the control efficiency, one may naturally wander how the "Local" algorithm performs on the network power efficiency. In this section, we vary different parameters such as the number of sensor nodes, the coverage ratio requirement, the transmission range to extensively check the power efficiency of our proposed algorithm. Before any dynamic event, "Original" results are obtained as references.



(a) On the overall sensing rate

(b) On the sensing duration

Fig. 4. Validation on the correctness and accuracy of Theorem 1



(a) Instant rescheduling time for 1000 time units

(b) CDF of the rescheduling time

Fig. 5. On the rescheduling time

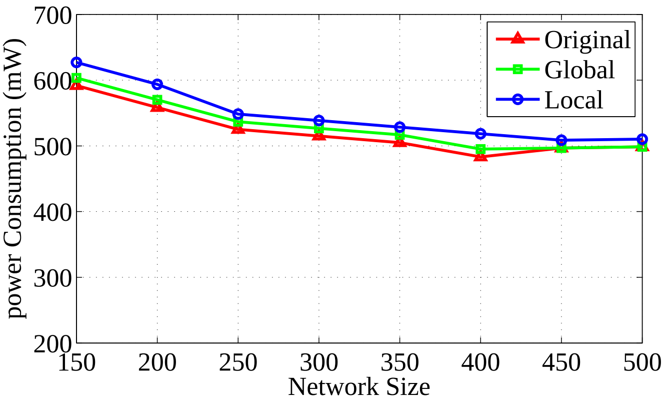


Fig. 6. On the Effect of Network Size

7.3.1 *The Effect of Network Size*

We first investigate the effect of network size $|S|$ under the values of $c_t = 1ms, f_t = 100Hz, \delta_t = 0.3, r_t = 6, \forall t \in T$. The number of sensors $|S|$ is varied from 150 to 500. Fig. 6 shows the results after sensor node vanishing event. For all algorithms, it can be first noticed that the power consumption first decreases and then slightly increases network size $|S|$. This is because when the network size is small, more sensor nodes imply more candidates to satisfy the quality-of-sensing requirement and thus less power will be consumed. However, when the network size is large enough to easily find the sensors that shall be activated for quality-of-sensing guaranteeing. Further increasing the network size does not benefit the power efficiency but introduces more non-negligible power consumption from sleeping nodes. Thus, the power consumption slightly increases. One may also observe that the gap between local and global shrinks with the increasing of network size. The reason is that there are more candidate sensors near the vanishing sensor node when the sensor node density is high and it is easy to find a substitution to the vanishing one. Actually, we find out that the "local" result sometimes is even the same as "global" one in some cases. In few cases, two or more sensor nodes are needed to compensate the vanishing of one sensor node with many sensing tasks. Nevertheless, under any network size, we

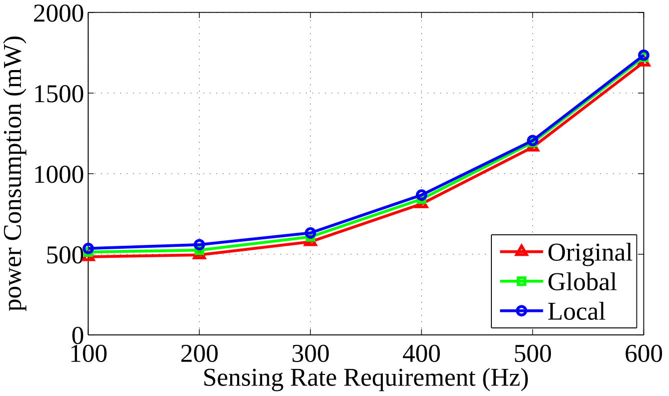


Fig. 7. On the Effect of Sensing Rate Requirement

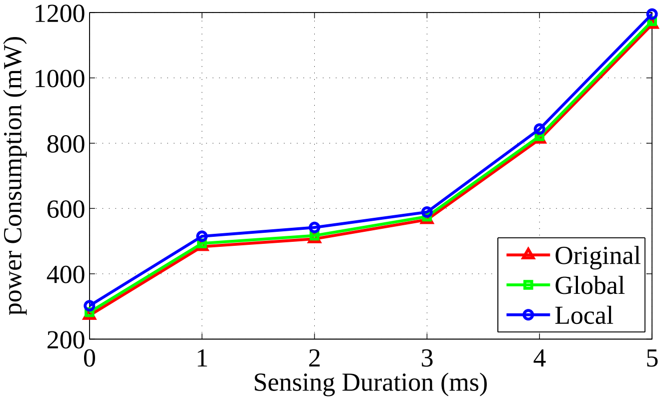


Fig. 8. On the Effect of Sensing Rate Requirement

can always see the closeness of our "local" algorithm to the optimal "global" one.

7.3.2 The Effect of Sensing Rate Requirement

According to Theorem 1, the sensing rate requirement also has deep influence on the sensor activation and hence the overall power consumption. To this end, we conduct a group of experiments under the settings $|S| = 400, c_t = 1ms, \delta_t = 0.3, r_t = 6, \forall t \in T$. The results are reported in Fig. 7, where the value of sensing rate requirement varies from 100Hz to 600Hz. We observe that the overall power consumption shows as a superlinear increasing function of the required sensing rate. When the rate is low, to activate a single sensor is sufficient to conduct the sensing tasks for many targets. When the rate is high, the number of activated sensors sharply increases in order to satisfy the tough sensing requirement.

7.3.3 The Effect of Sensing Duration

Besides the sensing rate requirement, the sensing duration also affects the power consumption. To this end, we also investigate the effect of sensing duration. The experimental results, where the duration varying from $1ms$ to $5ms$, are shown in Fig. 8 under the setting $|S| = 400, f_t = 100Hz, \delta_t = 0.3, r_t = 6, \forall t \in T$. We notice that the power consumption is a superlinear function of the sensing duration for the similar reason.

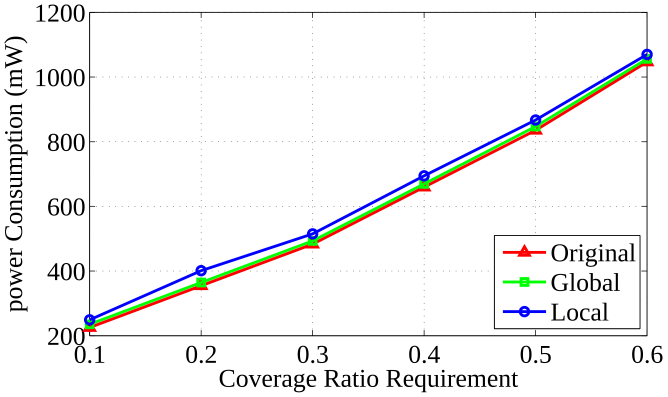


Fig. 9. On the Effect of Coverage Ratio Requirement

7.3.4 The Effect of Coverage Ratio Requirement

Next, we check how the quality-of-sensing requirement affects the network power efficiency. Fig. 9 shows the results under the values of $|S| = 500, c_t = 1ms, f_t = 100Hz, r_t = 6, \forall t \in T$. The coverage ratio requirement for all tasks is increased from 0.1 to 0.6. We find out that the power consumption almost linearly increases with the values of δ for all algorithms. This is due to the fact that more sensors shall be activated to cover more targets to satisfy the required quantity-of-sensing. For example, according to the "original" results, around 32 nodes shall be activated when the coverage ratio is set as 0.3 while the value increases to 70 when the requirement becomes 0.6. After a dynamic event, "global" achieves almost the same performance as "original" while "local" requires little higher power as only local information is exploited. For example, as shown in Fig. 3, after the vanishing of node 10, node 7 can be activated to ensure the coverage if global optimization is applied, other than activating both nodes 8 and 9 by local optimization. However, remember that global optimization is at the expense of high scheduling time and control overhead.

7.3.5 The Effect of Sensing Range

We finally check the effect from the sensing range, which determines the reachability of sensor nodes to the targets. We investigate its effect via setting $|S| = 500, c_t = 1ms, f_t = 100Hz, \delta_t = 0.3, \forall t \in T$. and vary the sensing range from 5m to 10m. The evaluation results are shown in Fig. 10, from which we can see that the power consumption shows as a decreasing function of the sensing range. This is consistent with our intuition that more targets can be reached by the sensor nodes with longer sensing duration and potentially smaller number of sensor nodes shall be activated to ensure the quality-of-sensing. However, we further notice that the decreasing rate becomes low with the increasing of sensing range. This is because reachability is not

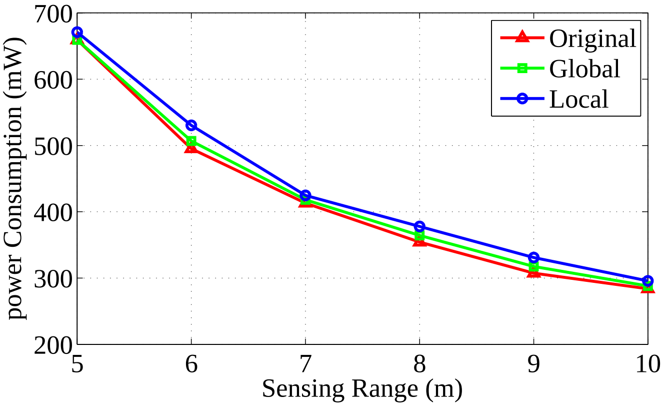


Fig. 10. On the Effect of Sensing Range

equivalent to coverage. Although more targets can be reached with long sensing range, the ability to cover these targets is still constrained by the schedulability and resource capacities. Thus, further increasing of sensing range does not take too much benefit to the power efficiency any more.

8 Conclusion

In this paper, we consider a minimum-energy activation and scheduling problem in multi-task SDSNs with quality-of-sensing guaranteed. We first formally derive the effective sensing rate that can be achieved by collaborative sensing from multiple sensors in closed-form. Based on our analysis, we build an MIQP formulation to describe the minimum-energy activation problem and then linearize it into MILP to lower the computation complexity. We further notice that it is unnecessary to always apply global optimization on the whole network. To this end, we further propose an online algorithm using local information near the dynamic event point to deal with the dynamic events that may happen during the SDSN runtime. Through extensive simulation studies, we prove the high efficiency our online algorithm using local optimization by the fact that it much approaches the network energy efficiency using global optimization but substantially outperforms it on rescheduling time. As part of our future work, network connectivity and communication energy consumption shall be taken into consideration.

References

[1] T. Luo, H.-P. Tan, and T. Quek, "Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[2] T. Miyazaki, D. Shitara, Y. Endo, Y. Tanno, H. Igari, and R. Kawano, "Die-hard Sensor Network: Robust Wireless Sensor Network Dedicated to Disaster Monitoring," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication (ICUIMC)*. ACM, 2011, pp. 53:1–53:10.

[3] T. Miyazaki, "Dynamic Function Alternation to Realize Robust Wireless Sensor Network," *International Journal of Handheld Computing Research (IJHCR)*, vol. 3, no. 3, pp. 17–34, Jul. 2012.

[4] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer *et al.*, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*. Springer, 2005, pp. 115–148.

[5] B. Wang, "Coverage Problems in Sensor Networks: A Survey," *ACM Computer Survey*, vol. 43, no. 4, pp. 32:1–32:53, Oct. 2011.

[6] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, "SYNAPSE++: Code Dissemination in Wireless Sensor Networks Using Fountain Codes," *IEEE Transactions on Mobile Computing*, vol. 9, no. 12, pp. 1749–1765, 2010.

[7] N. Bin Shafi, K. Ali, and H. Hassanein, "No-reboot and Zero-Flash Over-the-air Programming for Wireless Sensor Networks," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 371–379.

[8] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the International Conference on Communications (ICC)*, vol. 2. IEEE, 2001, pp. 472–476.

[9] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.

[10] C.-F. Huang and Y.-C. Tseng, "The Coverage Problem in a Wireless Sensor Network," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*. ACM, 2003, pp. 115–121.

[11] M. X. Cheng, L. Ruan, and W. Wu, "Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 4. IEEE, 2005, pp. 2638–2645.

[12] K. Kar, A. Krishnamurthy, and N. Jaggi, "Dynamic node activation in networks of rechargeable sensors," in *Proceedings fo the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3. IEEE, 2005, pp. 1997–2007.

[13] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and Coverage Guarantees Through Distributed Coordinate-Free Sensor Activation," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 470–483, 2011.

[14] C.-f. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04. New York, NY, USA: ACM, 2004, pp. 433–442.

[15] S. He, J. Chen, D. K. Yau, H. Shao, and Y. Sun, "Energy-efficient capture of stochastic events by global- and local-periodic network coverage," in *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '09. New York, NY, USA: ACM, 2009, pp. 155–164.

[16] S. He, J. Chen, and Y. Sun, "Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 475–482, 2012.

[17] Y. Tian, E. Ekici, and F. Ozguner, "Energy-constrained task mapping and scheduling in wireless sensor networks," in *Proceedings of the International Conference on Mobile Adhoc and Sensor Systems (MASS)*. IEEE, 2005.

[18] A. Voinescu, D. Tudose, and N. Tapus, "Task Scheduling in Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Networking and Services (ICNS)*, 2010, pp. 12–17.

[19] S. Habib and P. Marimuthu, "Data aggregation at the gateways through sensors' tasks scheduling in wireless sensor networks," *IET Wireless Sensor Systems*, vol. 1, no. 3, pp. 171–178, 2011.

[20] N. Edalat, C.-K. Tham, and W. Xiao, "An auction-based strategy for distributed task allocation in wireless sensor networks," *Computer Communications*, vol. 35, no. 8, pp. 916–928, 2012.

[21] W. Li, F. C. Delicato, and A. Y. Zomaya, "Adaptive energy-efficient scheduling for hierarchical wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 3, pp. 33:1–33:34, Jun. 2013.

[22] W. Li, F. C. Delicato, P. F. Pires, Y. C. Lee, A. Y. Zomaya, C. Miceli, and L. Pirmez, "Efficient allocation of resources in multiple heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1775–1788, 2014.

[23] C. M. de Farias, L. Pirmez, F. C. Delicato, W. Li, A. Y. Zomaya, and J. N. de Souza, "A scheduling algorithm for shared sensor

and actuator networks," in *Information Networking (ICOIN), 2013 International Conference on*. IEEE, 2013, pp. 648–653.

[24] Y. Xu, A. Saifullah, Y. Chen, C. Lu, and S. Bhattacharya, "Near optimal multi-application allocation in shared sensor networks," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010, pp. 181–190.

[25] N. Kapoor, S. Majumdar, and B. Nandy, "System and application knowledge based scheduling of multiple applications in a wsn," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 350–355.

[26] S. Tang and J. Yuan, "Damson: On distributed sensing scheduling to achieve high quality of monitoring," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 155–159.

[27] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[28] J. Wang and N. Zhong, "Efficient point coverage in wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 291–304, 2006.

[29] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Sensor placement with multiple objectives for structural health monitoring," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 4, p. 68, 2014.

[30] D. K. Yau, N. K. Yip, C. Y. Ma, N. S. Rao, and M. Shankar, "Quality of monitoring of stochastic events by periodic and proportional-share scheduling of sensor coverage," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 2, p. 18, 2010.

[31] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on*. IEEE, 2002, pp. 129–136.

[32] R. Willett, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 124–133.

[33] J. Chen, K. Cao, K. Li, and Y. Sun, "Distributed sensor activation algorithm for target tracking with binary sensor networks," *Cluster Computing*, vol. 14, no. 1, pp. 55–64, 2011.

[34] N. Jaggi, S. Madakasira, S. Reddy Mereddy, and R. Pendse, "Adaptive algorithms for sensor activation in renewable energy based sensor systems," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1405–1420, 2013.

[35] J.-M. Garcia, O. Brun, and D. Gauchard, "Transient analytical solution of m/d/1/n queues," *Journal of applied probability*, pp. 853–864, 2002.

[36] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, "On task schedulability in real-time control systems," in *Proceedings of the 17th Real-Time Systems Symposium (RTSS)*. IEEE, 1996, pp. 13–21.

[37] D. Bienstock, "Computational study of a family of mixed-integer quadratic programming problems," *Mathematical Programming*, vol. 74, no. 2, pp. 121–140, 1996.

[38] Gurobi Optimization, "Gurobi Optimizer Reference Manual," 2013. [Online]. Available: http://www.gurobi.com

PLACE
PHOTO
HERE

Deze Zeng received his Ph.D. and M.S. degrees in computer science from University of Aizu, Aizu-Wakamatsu, Japan, in 2013 and 2009, respectively. He is currently a research assistant in University of Aizu, Japan. He received his B.S. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2007. He is a member of IEEE. His current research interests include cloud computing, networking protocol design and analysis, with a special emphasis on delay-tolerant networks and wireless sensor networks.

PLACE
PHOTO
HERE

Peng Li received his BS degree from Huazhong University of Science and Technology, China, in 2007, the MS and PhD degrees from the University of Aizu, Japan, in 2009 and 2012, respectively. He is currently a Postdoctoral Researcher in the University of Aizu, Japan. His research interests include networking modeling, cross-layer optimization, network coding, cooperative communications, cloud computing, smart grid, performance evaluation of wireless and mobile networks for reliable, energy-efficient, and cost-effective communications.

PLACE
PHOTO
HERE

Song Guo received the PhD degree in computer science from the University of Ottawa, Canada in 2005. He is currently a Full Professor at School of Computer Science and Engineering, the University of Aizu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for reliable, energy-efficient, and cost effective communications in wireless networks. He has published over 250 papers in refereed journals and conferences in these areas and received three IEEE/ACM best paper awards. Dr. Guo currently serves as Associate Editor of IEEE Transactions on Parallel and Distributed Systems, Associate Editor of IEEE Transactions on Emerging Topics in Computing with duties on emerging paradigms in computational communication systems, and on editorial boards of many others. He has also been in organizing and technical committees of numerous international conferences. Dr. Guo is a senior member of the IEEE and the ACM.

PLACE
PHOTO
HERE

Toshiaki Miyazaki is a Professor of the University of Aizu, Fukushima, Japan, and the Chair of Department of Computer and Information Systems, Graduate School of the University of Aizu.. He is the Director of Research Center for Advanced Information Science and Technology (CAIST), the University of Aizu. He is also the Deputy Director of the Revitalization Center, the University of Aizu. His research interests are in reconfigurable hardware systems, adaptive networking technologies, and autonomous systems. He received the B.E. and M.E. degrees in applied electronic engineering from the University of Electro-Communications, Tokyo, Japan in 1981 and 1983, and Ph.D. degree in electronic engineering from Tokyo Institute of Technology in 1994. Before joining the University of Aizu, he has worked for NTT for 22 years, and engaged in research on VLSI CAD systems, telecommunications-oriented FPGAs and their applications, active networks, peer-to-peer communications, and ubiquitous network environments. Dr. Miyazaki was a visiting professor of the graduate school, Niigata University in 2004, and a part-time lecturer of the Tokyo University of Agriculture and Technology in 2003-2007. He is a senior member of IEEE, and a member of IEICE and IPSJ.

PLACE
PHOTO
HERE

Jiankun Hu is Professor and Research Director of Cyber Security Lab, School of Engineering and IT, University of New South Wales at the Australian Defence Force Academy (UNSW@ADFA), Canberra, Australia. He has obtained his BE from Hunan University, China in 1983; PhD in Control Engineering from Harbin Institute of Technology, China in 1993 and Masters by Research in Computer Science and Software Engineering from Monash University, Australia in 2000. He has worked in Ruhr University Germany on the prestigious German Alexander von Humboldt Fellowship 1995-1996; research fellow in Delft University of the Netherlands 1997-1998, and research fellow in Melbourne University, Australia 1998-1999. Jiankun's main research interest is in the field of cyber security including biometrics security where he has published many papers in high-quality conferences and journals including IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). He has served in the editorial board of up to 7 international journals and served as Security Symposium Chair of IEEE flagship conferences of IEEE ICC and IEEE Globecom. He has obtained 7 ARC (Australian Research Council) Grants and is now serving at the prestigious Panel of Mathematics, Information and Computing Sciences (MIC), ARC ERA (The Excellence in Research for Australia) Evaluation Committee.

PLACE
PHOTO
HERE

Yong Xiang received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1983 and 1989, respectively, and the Ph.D. degree from The University of Melbourne, Melbourne, Australia, in 2003. He was with the Southwest Institute of Electronic Equipment of China, Chengdu, from 1983 to 1986. He joined the University of Electronic Science and Technology of China in 1989, where he was a Lecturer from 1989 to 1992, and an Associate Professor from 1992 to 1997. He was a Senior Communications Engineer with Bandspeed Inc., Melbourne, Australia, from 2000 to 2002. He is currently an Associate Professor with the School of Information Technology, Deakin University, Melbourne, Australia. His current research interests include blind signal and system estimation, communication signal processing, biomedical signal processing, information and network security, and speech and image processing.