

Virtually Moving Base Stations for Energy Efficiency in Wireless Sensor Networks

Runwei Zhang
I&C, EPFL
Lausanne, Switzerland
runwei.zhang@epfl.ch

Patrick Thiran
I&C, EPFL
Lausanne, Switzerland
patrick.thiran@epfl.ch

Martin Vetterli
I&C, EPFL
Lausanne, Switzerland
martin.vetterli@epfl.ch

ABSTRACT

Energy efficiency of wireless sensor networks (WSNs) can be improved by moving base stations (BSs), as this scheme evenly distributes the communication load in the network. However, physically moving the BSs is complicated and costly. In this paper, we propose a new scheme: *virtually moving the BSs*. We deploy an excessive number of BSs and adaptively re-select a subset of active BSs so as to emulate the physical movement. Beyond achieving high energy-efficiency, this scheme obviates the difficulties associated with physically moving the BSs.

The challenges are (i) that the energy efficiency of BSs should be considered as well, in addition to that of the sensor nodes and (ii) that the number of candidate subset of active BSs is exponential with the number of BSs. We show that scheduling the virtual movement of BSs is NP-hard. Then, we propose a polynomial-time algorithm that is guaranteed under mild conditions to achieve a lifetime longer than 62% of the optimal one. In practice, as verified through extensive numerical simulations, the lifetime achieved by the proposed algorithm is always very close to the optimum.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication, Distributed networks; C.2.2 [Network Protocols]: Routing protocols

General Terms

Theory

Keywords

Wireless sensor networks, moving base stations, energy efficiency, load balancing

1. INTRODUCTION

Energy efficiency is a key issue in wireless sensor networks (WSNs), because the batteries of sensor nodes have limited

capacity. Specifically-designed routing protocols [21, 7, 1] can, to some extent, improve the energy efficiency of the WSN. Yet, they can do little for the *energy-hole* problem: sensor nodes close to the base stations (BSs)¹ forward much more data and drain their batteries much faster than other sensor nodes. The uneven distribution of the energy consumption is a culprit for the poor energy efficiency.

Moving BSs [13, 18, 9, 3, 14] addresses the energy-hole problem, as it increases the degree of freedom for evenly distributing the energy consumption in the WSN. Properly scheduled movements of BSs can greatly improve the energy efficiency of the WSN. However, all existing approaches exploiting BS mobility have a major drawback: they make BSs physically move, which requires BSs to have additional implements for mobility. As a result, practical implementations of these approaches are complicated and costly.

Can we reap the benefits of moving BSs, while avoiding the hassles of the physical movement? We give an affirmative answer by proposing a new scheme: *virtually moving the BSs*. To emulate the physical movement, we deploy an excessive number of BSs in the sensing field and continue re-selecting a subset of active BSs. Active BSs turn on long-range communication and forward data to the remote server; and passive BSs turn off the long-range communication and perform like sensor nodes. Simply by switching on and off the long-range communication, we balance the energy-consumption load of all sensor nodes and all BSs. In this scheme, BSs are also deployed in the sensing field and are supplied with limited available energy. Therefore, keeping all deployed BSs always active is not energy efficient and we have to optimize both the energy efficiency of BSs and that of sensor nodes. We term this joint optimization problem a *virtually-moving BSs problem*. This problem is much more complicated than the traditional maximum lifetime routing problem (e.g., [7]), because active BSs consume a large and fixed amount of energy for using long-range communications. Based on our measurement through a digital oscilloscope, a typical GSM/GPRS module on BSs has to be activated for around 35 seconds just to start and close each long-range communication connection. In the meantime, depending on the sensor sampling rate of the WSN, it only takes a few seconds for actually transmitting the sensing samples. Therefore, BSs can consume energy an order of magnitude more on the fixed energy-consumption than on actually transmitting the sensing samples. This fixed energy-consumption

¹BSs are the devices that forward data from all sensor nodes across the WSN to a remote server through long-range communication, e.g., GSM/GPRS, EDGE, 3GPP or LTE.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MobiHoc '15, June 22–25, 2015, Hangzhou, China.
Copyright © 2015 ACM 978-1-4503-3489-1/15/06 ...\$15.00.
<http://dx.doi.org/10.1145/2746285.2746291>.

of BSs makes the virtually-moving problem NP-hard, as we will present a reduction from the 3-SAT [6].

The virtually-moving BSs problem cannot be solved using the traditional Garg-Konemann typed algorithms [10], as these algorithms only deal with linear packing problems. We propose a novel scheme that transform the problem into a sequence of *uncapacitated facility-location* (UFL) problems [12] by using the constrained gradient method [8]. Under very mild conditions, the obtained UFL problems have approximate solutions with an approximation ratio of 1.61 [12]. By adaptively solving the sequence of UFL problems, our proposed algorithm yields a lifetime at least 62% of the optimum. The proposed algorithm is adaptive, has low computational complexity and requires easily available information as input. It is therefore a perfect fit for the WSN paradigm.

Through extensive simulations, we show that virtually moving multiple BSs is more energy efficient compared to other existing schemes for organizing WSNs. We also show that the achieved lifetime of the proposed scheduling algorithm is always close to the optimum.

The main contributions of this paper are as follows:

1. We propose a new scheme for organizing WSNs which virtually moves the BSs. This scheme not only achieves high energy-efficiency, but it also circumvents the challenges of physically moving BSs. The idea of load balancing by virtually moving devices can be used in other networked systems, including super-node selection in peer-to-peer networks and cooperative beamforming in cellular networks.
2. We show that the scheduling problem of virtually moving BSs is NP-hard, and we propose an adaptive algorithm to solve it. The proposed algorithm is computationally light, only requires easily available information as input and guarantees, under mild conditions, to yield a lifetime that is at least 62% of the optimal one. Due to the general formulation, the proposed scheduling algorithm has merits beyond the WSNs.

The outline of this paper is as follows. First, we review the related works in Section 2. Then, we show the system model in Section 3 and formally formulate the scheduling problem of virtually moving BSs in Section 4. Next, we analyze the complexity of the scheduling problem in Section 5 and propose the adaptive algorithm in Section 6. Finally, we show simulation results in Section 7 and conclude in Section 8.

2. RELATED WORK

Energy efficiency is a key issue in the design of WSNs. Since the emergence of WSNs, researchers have proposed many schemes to enhance the energy efficiency of WSNs.

A traditional WSN has a single BS and a number of sensor nodes deployed around the BS. At the very beginning, researchers have proposed to optimize the multi-hop routing of the short-range communication of sensor nodes [21, 7, 1]. These schemes, to some extent, improve the energy efficiency of the WSN. Yet, they can do little for the energy-hole problem: sensor nodes near the single BS deplete their batteries much faster than other sensor nodes. The uneven distribution of the energy consumption greatly undermines the energy efficiency of the network.

To mitigate the energy-hole problem, researchers propose to deploy multiple BSs [5, 11, 17]: Instead of using one single

BS, they set up multiple BSs and let sensor nodes around all BSs share the high communication load. By optimizing the positions of the multiple BSs, the energy-consumption load is more evenly balanced, resulting in a higher energy efficiency.

To completely solve the energy-hole problem, researchers propose to physically move BSs [13, 4, 15, 9, 16, 2, 14]. These schemes have high energy-efficiency because they increase the degree of freedom for evenly distributing the communication load. However, optimizing the continuous moving trace of BSs is hard. Even in WSNs with one BS, the optimal trace of the BS can only be found when the sensing field has specific geometric shapes, such as, circles [13]. Then, researchers propose to discretize the continuous moving traces of BSs by restricting the positions of BSs to a set of candidate positions. Shi *et al.* [15] show that such a discretization incurs an arbitrarily small loss of precision by constructively selecting the candidate positions. Then, Wang *et al.* [2] discusses the scheduling problem for moving a single BS on a finite set of candidate positions. The recent work of Luo *et al.* [14] discuss the problem of moving a given number of BSs on a finite set of candidate positions. They show that this problem is NP-hard and present an approximation algorithm for scheduling the movements.

Because physically moving the BSs is complicated and costly, we propose a new scheme that virtualizes this movement. It offers two additional differences with physical motion of the BSs (*e.g.*, [2, 14]): (i) In the virtual-movement scheme, the number of active BSs selected at different times can be changed, contrary to the physical-movement scheme, and (ii) in the virtual-movement scheme, BSs are considered to have a finite energy supply like sensor nodes because BSs are also installed in the sensing field; whereas in the physical-movement scheme, BSs are, to the best of our knowledge, always assumed to have an infinite energy supply. In our recent work [19], we discussed virtually moving one active BS and implemented this scheme in a real WSN. In this paper, we consider the more general scheme where multiple BSs are virtually moving.

3. SYSTEM MODEL

We model the WSN as a directed graph whose vertices are a set of BSs V_b , a set of sensor nodes V_r and a remote server S . The edges in the directed graph consist of (i) short-range communication links among all BSs and sensor nodes $V_b \cup V_r$, and (ii) long-range communication links from BSs V_b to the remote server S . We now introduce the definitions for data communication and energy consumption in the considered WSN. The notations are summarized in Table 1.

3.1 Data Communication

Each sensor node $i \in V_r$ generates data with a constant rate $r \in \mathbb{R}^+$, whereas BSs do not generate data. All data has to be transmitted to active BSs by short-range multi-hop communications. Active BSs forward the collected data to the remote server S via long-range communication. Although active BSs are adaptively changed, sensor nodes are always actively sensing and uploading data to the remote server via active BSs. Let $x_{ij} \in \mathbb{R}_0^+$ be the data rate sent on the short-range communication link from i to j ($i, j \in V_b \cup V_r$). Let $y_i \in \mathbb{R}_0^+$ be the data rate sent on the long-range communication link from BS $i \in V_b$ to the remote server S . Because the rate of data inflow and outflow should

Table 1: Notations

r	data generating rate of each sensor node
x_{ij}	data rate on short-range communication link from i to j
$\mathbf{x} (\mathbf{x}^{(n)})$	data rates on all short-range communication links (at time slot n)
y_i	data rate on long-range communication link of BS j
$\mathbf{y} (\mathbf{y}^{(n)})$	data rates on long-range communication link of all BSs (at time slot n)
z_i	indicator activity state of BS i
$\mathbf{z} (\mathbf{z}^{(n)})$	vector of indicators showing the active states of all BSs (at time slot n)
c_i	energy consumption rate of node i
$\mathbf{c} (\mathbf{c}^{(n)})$	energy consumption rates of all sensor nodes and BSs (at time slot n)
$c_c, c_{st}, c_{sr}, c_{lc}, c_{lt}$	constant parameters for modelling the energy consumption rates
e_i	initially available energy of node i
$\mathbf{e} (\mathbf{e}^{(n)})$	initially available energy of all sensor nodes or BSs (at time slot n)
\mathcal{L}	the set of all candidate configurations
$\mathcal{L}^{[k]}$	the set of candidate configurations given that $\mathbf{z} = \mathbf{z}^{[k]}, 1 \leq k \leq 2^{ V_b } - 1$
θ_i	average energy decrease rate of node i
$\boldsymbol{\theta} (\boldsymbol{\theta}^{(n)})$	average energy decrease rate of all sensor nodes or BSs (at time slot n)
T^*	the optimal lifetime of the virtually-moving BSs problem (5)
τ	length of a time slot
N^*	number of time slots that the WSN can sustain when using Algorithm 1

be balanced both on sensor nodes and on BSs, we have

$$\left\{ \begin{array}{l} \sum_{j \in V_b \cup V_r} x_{ij} = \sum_{j \in V_b \cup V_r} x_{ji} + r, \forall i \in V_r, \\ \sum_{j \in V_b \cup V_r} x_{ij} + y_i = \sum_{j \in V_b \cup V_r} x_{ji}, \forall i \in V_b, \\ x_{ij} \geq 0, i, j \in V_b \cup V_r, \\ y_i \geq 0, i \in V_b. \end{array} \right. \quad (1)$$

Let z_i ($i \in V_b$) be a binary variable indicating the state of BS i : if BS i is active, $z_i = 1$, and otherwise $z_i = 0$. When BS i is passive, its data rate of long-range communication y_i should be 0. Therefore,

$$y_i \leq |V_r| r z_i, \quad (2)$$

where $|V_r| r$ is the total data rate transmitted in the whole WSN. If BS i is active, (2) imposes no constraint on y_i .

3.2 Energy Consumption

We denote the energy consumption rate of either a sensor node or a BS $i \in V_b \cup V_r$ by c_i ($c_i \in \mathbb{R}^+$). (i) On the one

hand, for a sensor node,

$$c_i = c_c + c_{st} \sum_{j \in V_b \cup V_r} x_{ij} + c_{sr} \sum_{j \in V_b \cup V_r} x_{ji}, \forall i \in V_r, \quad (3)$$

where c_c denotes the constant energy consumption rate for sensing and initiating the short-range communication, and c_{st}, c_{sr} denote the energy consumption rates for transmitting and receiving unit data via short-range communication, respectively. (ii) On the other hand, for a BS,

$$c_i = c_c + c_{st} \sum_{j \in V_b \cup V_r} x_{ij} + c_{sr} \sum_{j \in V_b \cup V_r} x_{ji} + c_{lc} z_i + c_{lt} y_i, \forall i \in V_b, \quad (4)$$

where c_{lc} denotes the fixed energy-consumption rate for initiating and closing long-range communication and c_{lt} denotes the energy consumption rate for actually transmitting unit data via long-range communication.

Let the initially available energy of each sensor node or BS i ($i \in V_b \cup V_r$) be denoted by e_i .

For simplicity of discussion, we group $\{x_{ij}\}_{i,j \in V_b \cup V_r}$ into a vector \mathbf{x} , group $\{y_i\}_{i \in V_b}$ into a vector \mathbf{y} , group $\{z_i\}_{i \in V_b}$ into a vector \mathbf{z} , group $\{c_i\}_{i \in V_b \cup V_r}$ into a vector \mathbf{c} and group $\{e_i\}_{i \in V_b \cup V_r}$ into a vector \mathbf{e} .

Notice that the state of a WSN is characterized by the data rates of short-range communication \mathbf{x} , the data rates of long-range communication \mathbf{y} , the selection of active BSs \mathbf{z} and the energy consumption rates \mathbf{c} . We call the four-tuple $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c})$ a *configuration* of the network.

4. PROBLEM FORMULATIONS

Scheduling the virtual movement of BSs is equivalent to finding a set of configurations and finding the time durations. In this section, we first describe the optimization space for selecting the configurations. Then, we will formally formulate the scheduling problem, namely, the *virtually-moving BSs problem*. At the end of this section, we will show why our proposed scheme is more energy efficient than the previously proposed schemes [5, 11, 17, 2, 19].

4.1 The Optimization Space

We denote the set of all candidate configurations by

$$\mathcal{L} = \{(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) | (1), (2), (3) \text{ and } (4) \text{ hold}\}.$$

Because \mathbf{z} is a binary vector taking $2^{|V_b|} - 1$ possible values (there should be at least one active BS, therefore we cannot have $\mathbf{z} = \mathbf{0}$), we can separate \mathcal{L} into $2^{|V_b|} - 1$ subsets based on different values of \mathbf{z} . We denote the possible values of \mathbf{z} by $\mathbf{z}^{[1]}, \mathbf{z}^{[2]}, \dots, \mathbf{z}^{[2^{|V_b|}-1]}$ with a non-decreasing order of the number of active BSs. In particular, the first $|V_b|$ elements, $\mathbf{z}^{[1]}, \mathbf{z}^{[2]}, \dots, \mathbf{z}^{[|V_b|]}$ denote the BSs selections where only one BS is active. For any $1 \leq k \leq 2^{|V_b|} - 1$, we denote by $\mathcal{L}^{[k]}$ the subset of \mathcal{L} satisfying $\mathbf{z} = \mathbf{z}^{[k]}$:

$$\mathcal{L}^{[k]} = \{(\mathbf{x}, \mathbf{y}, \mathbf{z}^{[k]}, \mathbf{c}) | (1), (2), (3), (4) \text{ hold}\},$$

which is a simplex because constraints (1), (2), (3), (4) are linear after fixing $\mathbf{z} = \mathbf{z}^{[k]}$. Decomposing $\mathcal{L} = \bigcup_{k=1}^{2^{|V_b|}-1} \mathcal{L}^{[k]}$ enables us to simplify the optimization problem, as will be explained in the following.

4.2 The Virtually-moving BSs Problem

Because $\mathcal{L}^{[k]}$ is a simplex for all $1 \leq k \leq 2^{|V_b|} - 1$, the convex combination of any two configurations in $\mathcal{L}^{[k]}$ is still in $\mathcal{L}^{[k]}$. Let $(\mathbf{x}^{[1]}, \mathbf{y}^{[1]}, \mathbf{z}^{[1]}, \mathbf{c}^{[1]}) \in \mathcal{L}^{[k]}$ and $(\mathbf{x}^{[2]}, \mathbf{y}^{[2]}, \mathbf{z}^{[2]}, \mathbf{c}^{[2]}) \in \mathcal{L}^{[k]}$ be used for time durations $t^{[1]}$ and $t^{[2]}$. Using these two configurations is equivalent to using their convex combination

$$\left(\frac{t^{[1]}\mathbf{x}^{[1]} + t^{[2]}\mathbf{x}^{[2]}}{t^{[1]} + t^{[2]}}, \frac{t^{[1]}\mathbf{y}^{[1]} + t^{[2]}\mathbf{y}^{[2]}}{t^{[1]} + t^{[2]}}, \mathbf{z}^{[k]}, \frac{t^{[1]}\mathbf{c}^{[1]} + t^{[2]}\mathbf{c}^{[2]}}{t^{[1]} + t^{[2]}} \right),$$

which is still in $\mathcal{L}^{[k]}$, for a time duration $t^{[1]} + t^{[2]}$. This further implies that any number of configurations in the same set $\mathcal{L}^{[k]}$ can be substituted by only one configuration—their convex combination.

Therefore, in the virtually-moving BSs problem, we seek at most $2^{|V_b|} - 1$ configurations: for each $1 \leq k \leq 2^{|V_b|} - 1$, we select a configuration $(\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$, such that the total lifetime² $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]}$ is maximized given the initially available energy e :

$$\begin{aligned} & \max_{\{\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}, t^{[k]}\}_{k=1}^{2^{|V_b|}-1}} \sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \\ & \text{s.t.} \quad \sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \mathbf{c}^{[k]} \leq e, \\ & \quad (\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}, \forall 1 \leq k \leq 2^{|V_b|} - 1, \\ & \quad t^{[k]} \geq 0, \forall 1 \leq k \leq 2^{|V_b|} - 1. \end{aligned} \quad (5)$$

We denote the optimal lifetime of (5) by T^* . To differentiate with other schemes that will be mentioned below, we call this scheme *MultiMove*, which schedules the virtual movement of multiple BSs by solving problem (5).

4.3 Comparisons to Other Schemes

In the following, we compare MultiMove with some other existing schemes.

MultiFixed [5, 11, 17]: This scheme selects a subset of BSs to be always active and selects the routes of short-range communications. This boils down to selecting only one configuration $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) \in \mathcal{L}$, so that the lifetime of the WSN is maximized:

$$\begin{aligned} & \max_{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}, t} t \\ & \text{s.t.} \quad (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) \in \mathcal{L}, \\ & \quad t\mathbf{c} \leq e, \end{aligned} \quad (6)$$

where t denotes the lifetime of the network, and $t\mathbf{c}$ denotes the total energy consumption that should be no larger than the initially available energy e .

OneMove [2, 19]: In this scheme, we schedule the virtual movement of one BS on a finite set of candidate locations for maximizing lifetime. This is equivalent to selecting at most $|V_b|$ configurations: for each $1 \leq k \leq |V_b|$ (in contrast to $1 \leq k \leq 2^{|V_b|} - 1$ as in the MultiMove scheme), we select one configuration $(\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}$ in each set $\mathcal{L}^{[k]}$ with the time duration $t^{[k]}$, such that the total lifetime

²There are many possible ways of defining the lifetime. Throughout this paper, we use the time that the first node drains out of energy as the lifetime of the WSN.

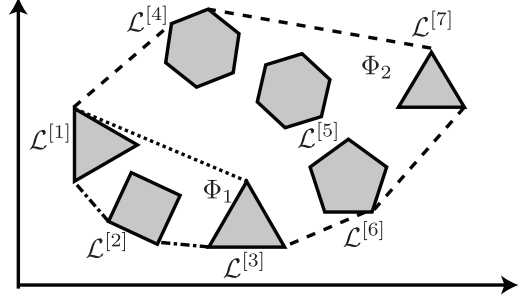


Figure 1: Optimization spaces of different schemes when the WSN has three BSs. The whole space for candidate configurations \mathcal{L} is the union of simplices $\mathcal{L}^{[1]}, \mathcal{L}^{[2]}, \dots, \mathcal{L}^{[7]}$ which are denoted by the grey polygons. The optimization spaces of MultiFixed is \mathcal{L} . The optimization space of OneMove is illustrated by the space $\Phi_1 = \text{conv}(\bigcup_{k=1}^3 \mathcal{L}^{[k]})$, whose edges are denoted by the dotted lines. The optimization space of MultiMove is illustrated by the space $\Phi_2 = \text{conv}(\mathcal{L})$, whose edges are denoted by the dashed lines. The common edges of Φ_1 and Φ_2 are denoted by dash-dot lines. We see that MultiMove has an optimization space larger than both MultiFixed and OneMove.

$\sum_{k=1}^{|V_b|} t^{[k]}$ is maximized:

$$\begin{aligned} & \max_{\{\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}, t^{[k]}\}_{k=1}^{|V_b|}} \sum_{k=1}^{|V_b|} t^{[k]} \\ & \text{s.t.} \quad \sum_{k=1}^{|V_b|} t^{[k]} \mathbf{c}^{[k]} \leq e, \\ & \quad (\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}, \forall 1 \leq k \leq |V_b|, \\ & \quad t^{[k]} \geq 0, \forall 1 \leq k \leq |V_b|. \end{aligned} \quad (7)$$

MultiMove is more energy efficient than the other schemes mentioned above. In the MultiFixed scheme, we select one configuration in the optimization space \mathcal{L} . In the MultiMove scheme, we select one configuration $(\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$ for any $1 \leq k \leq 2^{|V_b|} - 1$. This virtually creates a new configuration whose average energy consumption rates are $\sum_{k=1}^{2^{|V_b|}-1} \mathbf{c}^{[k]} t^{[k]} / \sum_{k=1}^{2^{|V_b|}-1} t^{[k]}$, a convex combination of $\mathbf{c}^{[1]}, \mathbf{c}^{[2]}, \dots, \mathbf{c}^{[2^{|V_b|}-1]}$. Therefore, the optimization space of MultiMove for selecting configurations is virtually expanded from \mathcal{L} to its convex hull $\text{conv}(\mathcal{L})$. Similarly, in the OneMove scheme, the optimization space is virtually expanded from $\bigcup_{k=1}^{|V_b|} \mathcal{L}^{[k]}$ (the subset of configurations where only one BS is active) to $\text{conv}(\bigcup_{k=1}^{|V_b|} \mathcal{L}^{[k]})$. We illustrate the optimization spaces of different schemes in Figure 1 with an example of a WSN with three BSs.

5. COMPLEXITY ANALYSIS

In this section, we will analyze the complexity of the virtually-moving BSs problem.

The problem is unsurprisingly “very hard” because (i) there is an exponential number of candidate subsets of active BSs, and (ii) active BSs have the fixed energy-consumption rate c_{1c} for using long-range communication, which makes the problem non-convex. To formally evaluate the hardness of the problem, we define the **decision version of**

the virtually-moving BSs problem as follows: Given the topology of the WSN, the constant parameters c_c , c_{st} , c_{sr} , c_{lc} , c_{lt} , the data generating rate r , the initially available energy e , the sets of possible configurations $\{\mathcal{L}^{[k]}\}_{k=1}^{2^{|V_b|}-1}$ defined by (1), (2), (3) and (4), and a number T , does there exist a configuration in each simplex $(\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}$ and a time duration $t^{[k]}$ for each $1 \leq k \leq 2^{|V_b|}-1$, such that the lifetime of the network $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \geq T$ under the energy constraint $\sum_{k=1}^{2^{|V_b|}-1} t^{[k]} \mathbf{c}^{[k]} \leq \mathbf{e}$?

THEOREM 1. *The virtually-moving BSs problem is NP-complete.*

PROOF. First of all, the decision version of the problem is NP because we can verify a valid instance in polynomial time. Now we show that the problem is NP-hard. We reduce the 3-SAT problem [6] to the virtually-moving BSs problem. Consider a 3-SAT instance with l variables a_1, a_2, \dots, a_l and m clauses b_1, b_2, \dots, b_m (Notice that each clause is a 3-element subset of $\{a_i, \bar{a}_i\}_{1 \leq i \leq l}$). We define an $2l + 1$ -th variable w other than $\{a_i, \bar{a}_i\}_{1 \leq i \leq l}$ and define a few sets:

- $P_i = \{a_i, \bar{a}_i\} \cup \{b_j | b_j \text{ contains } a_i\}, \forall 1 \leq i \leq l,$
- $Q_i = \{a_i, \bar{a}_i\} \cup \{b_j | b_j \text{ contains } \bar{a}_i\}, \forall 1 \leq i \leq l,$
- $W_1 = \{w\} \cup \{a_1, a_2, \dots, a_l\} \cup \{b_1, b_2, \dots, b_m\},$
- $W_2 = \{w\} \cup \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_l\}.$

We construct an instance of the virtually-moving BSs problem as follows. Let $P_1, P_2, \dots, P_l, Q_1, Q_2, \dots, Q_l, W_1, W_2$ be associated with a BS each, and let $a_1, a_2, \dots, a_l, b_1, b_2, \dots, b_m, w$ be associated with a sensor node each. Let each BS have a long-range communication link to the remote server S . Excluding the remote server S , the graph of the WSN is bipartite where every edge connects a BS to a sensor node, and this edge only exists when the set associated with the BS contains the variable associated with the sensor node. Let the initially available energy of each BS be 1 and let that of each sensor node be 2. Let the constants for energy consumption be $c_c = 0, c_{st} = 1, c_{sr} = 0, c_{lc} = 1, c_{lt} = 0$, and let the data generating rate $r = 1$. Then, we set the tentative lifetime $T = 2$.

Because the data generating rate $r = 1, c_{st} = 1$ and because of (3), the energy consumption rate of any sensor node is at least 1. Remember that the initially available energy of any sensor node is 2, the lifetime of the considered virtually-moving BSs problem is at most $T = 2$. In the following, we will show that if the constructed virtually-moving BSs problem achieves the lifetime $T = 2$, the original 3-SAT problem is satisfiable. We start with four deductions, given that the lifetime $T = 2$ is achievable:

(i) To be active, BSs W_1 and W_2 alternate, and each for a time duration 1. First, because of (4) and because $c_{lt} = 1$, the energy consumption rate of any BS is at least 1, therefore any BS has an active time duration at most 1 given the initially available energy 1. Then, because BSs W_1 and W_2 are the only neighbours of sensor node w and because they have to serve w for a whole lifetime 2, both BSs W_1 and W_2 have to be active for a time duration 1 and they cannot be simultaneously active.

(ii) The energy consumption rate of every sensor node is 1 at any time. On the one hand, it is at least 1, as shown

above. On the other hand, it cannot exceed 1, because every sensor node has to sustain a lifetime 2 with an initially available energy 2.

(iii) Passive BSs do not forward data for other sensor nodes at any time, otherwise, at least one sensor node would receive data from passive BSs. Using (3), we see that this would result in an energy consumption rate exceeding 1, which would contradict (ii).

(iv) In the configuration where BS W_2 is active (BS W_1 is passive because of (i)), one and only one BS between BS P_i and BS Q_i is active for any $1 \leq i \leq l$. First, at least one of them has to be active in order to forward data from sensor node a_i because they are the only neighbours of a_i . Then, they cannot be simultaneously active, because they have to forward data for sensor node a_i for a lifetime 2 with the total available energy 2 and because $c_{lt} = 1$.

Because of deduction (i), we have a configuration in which BS W_2 is active and BS W_1 is passive. We will use this configuration to construct a valid assignment for the original 3-SAT problem. Because of deduction (iv), one and only one BS between BSs P_i and Q_i is active for any $1 \leq i \leq l$ in that configuration. This enables us to construct an assignment for the 3-SAT problem by setting the variable $a_i = 1$ if BS P_i is active and setting $a_i = 0$ if BS Q_i is active. Then, because of deduction (iii), each sensor node in b_1, b_2, \dots, b_m connects to at least one active BS because passive BSs do not forward data from sensor nodes. Therefore, every clause b_j ($1 \leq j \leq m$) in the 3-SAT problem is satisfied: If sensor node b_j connects to the active BS P_i , clause b_j is satisfied because it contains variable a_i and $a_i = 1$; if sensor node b_j connects to the active BS Q_i , clause b_j is also satisfied because it contains variable \bar{a}_i and $a_i = 0$. In summary, if we can solve the constructed virtually-moving BSs problem, we can also solve the original 3-SAT problem. Therefore, the virtually-moving BSs problem is NP-hard as 3-SAT is known to be NP-hard. \square

6. SCHEDULING ALGORITHM

In this section, we will propose a scheme for adaptively scheduling the virtual movement of multiple BSs. The proposed algorithm is computationally light, only requires easily available information as input, and guarantees, under mild conditions, a network lifetime at least 62% of the optimal one.

The adaptive scheduling scheme works as follows. Time is discretized into slots of length τ , during which we use only one configuration. Before the start of each time slot, the remote server collects the information about the current available energy of all BSs and all sensor nodes. Using only this information as input, the remote server selects the configuration to be used in the next time slot and notifies all BSs. As τ is usually much larger than the sampling interval of sensor nodes, this scheme has negligible overhead for collecting the required information and distributing the configuration determined by the server.

We denote the available energy of all sensor nodes and BSs at time n ($n \in \mathbb{N}$) by $\mathbf{e}^{(n)}$. In particular, the initially available energy $\mathbf{e}^{(0)} = \mathbf{e}$. Denote the configuration selected for time slot n by $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)}, \mathbf{c}^{(n)}) \in \mathcal{L}$. The available energy evolves according to

$$\mathbf{e}^{(n)} = \mathbf{e}^{(n-1)} - \tau \mathbf{c}^{(n)}. \quad (8)$$

If we sum up the iterative equation (8) from time 0 to time n and use $\mathbf{e}^{(0)} = \mathbf{e}$, we have

$$\mathbf{e}^{(n)} = \mathbf{e} - \tau \sum_{t=1}^n \mathbf{c}^{(t)}. \quad (9)$$

Denote by N^* the maximum number of time slots before the first sensor node depletes its energy,

$$N^* = \max\{n | \mathbf{e}^{(n)} \geq \mathbf{0}\}, \quad (10)$$

and the lifetime of the WSN is τN^* .

In the following, to find the optimal configurations, we first propose to use the constrained gradient method [8] and to transform the virtually-moving BSs problem into adaptively solving a sequence of subproblem called the *min-weight configuration problems*. Then, we will discuss how to solve the min-weight configuration problems.

6.1 The Constrained Gradient Method

First of all, we define auxiliary variables

$$p^{[k]} = t^{[k]} / \sum_{k=1}^{2^{|V_b|}-1} t^{[k]}, \quad (11)$$

$$\boldsymbol{\theta} = \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} \mathbf{c}^{[k]}. \quad (12)$$

Here $p^{[k]}$ denotes the fraction of time for using configuration $(\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]})$ for all $1 \leq k \leq 2^{|V_b|} - 1$ and $\boldsymbol{\theta}$ denotes the average energy decrease rates for all sensor nodes and BSs. Next, we recast problem (5) as the auxiliary problem

$$\begin{aligned} \min_{\{\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}, p^{[k]}\}_{k=1}^{2^{|V_b|}-1}, \boldsymbol{\theta}} \quad & F(\boldsymbol{\theta}) = \max_{i \in V_b \cup V_r} \theta_i / e_i \\ \text{s.t.} \quad & \boldsymbol{\theta} = \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} \mathbf{c}^{[k]}, \\ & (\mathbf{x}^{[k]}, \mathbf{y}^{[k]}, \mathbf{z}^{[k]}, \mathbf{c}^{[k]}) \in \mathcal{L}^{[k]}, \forall 1 \leq k \leq 2^{|V_b|} - 1, \\ & \sum_{k=1}^{2^{|V_b|}-1} p^{[k]} = 1, \\ & p^{[k]} \geq 0, \forall 1 \leq k \leq 2^{|V_b|} - 1, \end{aligned} \quad (13)$$

where θ_i is the i -th element of $\boldsymbol{\theta}$. Note that the objective value of (13) is the inverse of the objective value in (5).

We relax the objective function $F(\boldsymbol{\theta})$ in problem (13) into a differentiable function

$$f(\boldsymbol{\theta}) = \frac{1}{\alpha} \log \left(\sum_{i \in V_b \cup V_r} \exp \left(\frac{\alpha \theta_i}{e_i} \right) \right) \quad (14)$$

by introducing a real parameter $\alpha > 0$. We can check that

$$f(\boldsymbol{\theta}) - \frac{1}{\alpha} \log |V_b \cup V_r| \leq F(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta}). \quad (15)$$

Therefore, using $f(\boldsymbol{\theta})$ as an approximation of $F(\boldsymbol{\theta})$ incurs an arbitrarily small loss of precision when α is large.

We define $\boldsymbol{\theta}^{(n)}$ as the average energy decrease rates during the first n time slots

$$\boldsymbol{\theta}^{(n)} = \frac{1}{n} \sum_{t=1}^n \mathbf{c}^{(t)} = \frac{\mathbf{e} - \mathbf{e}^{(n)}}{n\tau}, \quad (16)$$

Algorithm 1: The constrained gradient method for selecting configurations

Input: The parameters $c_c, c_{st}, c_{sr}, c_{lc}, c_{lt}, r$, the initially available energy \mathbf{e} , and a β -approximate algorithm for solving the min-weight configuration problem (18).

Output: N^* , $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)}, \mathbf{c}^{(n)})\}_{1 \leq n \leq N^*}$.

- 1 Initialize $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$, $\mathbf{e}^{(0)} \leftarrow \mathbf{e}$, $n \leftarrow 1$.
 - 2 **while** $\mathbf{e}^{(n-1)} \geq \mathbf{0}$ **do**
 - 3 Calculate $\boldsymbol{\theta}^{(n-1)}$ from $\mathbf{e}^{(n-1)}$ and (16).
 - 4 Calculate the gradient $\nabla^\top f(\boldsymbol{\theta}^{(n-1)})$ using (17).
 - 5 Select configuration $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)}, \mathbf{c}^{(n)})$ by using a β -approximate algorithm to solve the min-weight configuration problem (18) where $\boldsymbol{\lambda} = \nabla^\top f(\boldsymbol{\theta}^{(n-1)})$.
 - 6 Update $\mathbf{e}^{(n)} \leftarrow \mathbf{e}^{(n-1)} - \mathbf{c}^{(n)}$.
 - 7 Update $n \leftarrow n + 1$.
 - 8 Set $N^* = n - 1$.
-

where the second equality follows from (9). The purpose is to adaptively select the configurations of the WSN such that the sequence $\{\boldsymbol{\theta}^{(n)}\}_{n \in \mathbb{N}^+}$ gradually approaches the desired $\boldsymbol{\theta}$ in (13). Note that $\boldsymbol{\theta}^{(n-1)}$ is known at time slot n because $\mathbf{e}^{(n-1)}$ is already known and because of (16).

As shown in Algorithm 1, we iteratively do the following in each time slot $n \in \mathbb{N}^+$ as long as $\mathbf{e}^{(n-1)} \geq \mathbf{0}$:

(i) First, we calculate the energy decrease rates in the first $n-1$ time slots $\boldsymbol{\theta}^{(n-1)}$ using $\mathbf{e}^{(n-1)}$ and using (16), and we calculate the gradient of the relaxed objective function $\nabla^\top f(\boldsymbol{\theta}^{(n-1)})$ whose i -th element ($i \in V_b \cup V_r$) is

$$\nabla_i^\top f(\boldsymbol{\theta}^{(n-1)}) = \frac{\exp(\alpha \theta_i^{(n-1)} / e_i)}{e_i \sum_{i \in V_b \cup V_r} \exp(\alpha \theta_i^{(n-1)} / e_i)}. \quad (17)$$

Here, e_i in (17) is the initially available energy of node i .

(ii) Then, we define a weight vector $\boldsymbol{\lambda} = \nabla^\top f(\boldsymbol{\theta}^{(n-1)})$ and select a configuration $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)}, \mathbf{c}^{(n)})$ by solving

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}} \quad & \boldsymbol{\lambda}^\top \mathbf{c} \\ \text{s.t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) \in \mathcal{L}, \end{aligned} \quad (18)$$

which is termed as the *min-weight configuration problem*.

When the algorithm terminates, the lifetime of the network is τN^* , as defined in (10). Because N^* is dependent on τ and α , we will use $N^*(\tau, \alpha)$ for N^* to show explicitly the dependencies in the following.

The min-weight configuration problem is still combinatorial because there could be an exponential number of candidate configurations in \mathcal{L} . Nonetheless, we will show in Theorem 2 that if the min-weight configuration problem has a β -approximate algorithm (β can be either a constant or a big-O function of $|V_b \cup V_r|$), the virtually-moving BSs problem also has a β -approximate algorithm.

THEOREM 2. *Using Algorithm 1 for selecting the configurations $\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{z}^{(n)}, \mathbf{c}^{(n)})\}_{n \in \mathbb{N}^+}$ with a β -approximation algorithm for solving the min-weight configuration problem (18), the achieved lifetime $\tau N^*(\tau, \alpha)$ satisfies*

$$\frac{\tau N^*(\tau, \alpha) + \tau}{T^*} > \frac{1}{\beta + \gamma_1(\alpha) + \gamma_2(\tau, \alpha)},$$

Table 2: Mapping from the min-weight configuration problem into an UFL problem

Facility set \mathcal{F}	BSs V_b
Open facility set \mathcal{F}_o	Active BSs $\{u \in V_b z_u = 1\}$
Customer set \mathcal{C}	Sensor nodes V_r
Facility cost h_u ($u \in \mathcal{F}$)	$\lambda_u c_{1c}$, $u \in V_b$
Service cost g_{uv} ($u \in \mathcal{F}, v \in \mathcal{C}$)	$L_{uv} + \lambda_v c_c$, where L_{uv} is the length of the shortest-path from sensor node $v \in V_r$ to S via BS $u \in V_b$ with the distance assignment (22).

where

$$\gamma_1(\alpha) = \frac{\beta T^* \ln |V_b \cup V_r|}{\alpha},$$

$$\gamma_2(\tau, \alpha) = \frac{\alpha \tau |V_b \cup V_r|^2 c_{\max}^3 T^* (\ln \frac{e_{\min}}{\tau c_{\max}} + 1)}{2e_{\min}^3},$$

with e_{\min} denoting the minimum element of e and c_{\max} denoting an upper-bound on the energy consumption rate of any BS or any sensor node

$$c_{\max} = c_c + c_{1c} + (c_{st} + c_{sr} + c_{1t}) |V_b \cup V_r| r.$$

If we set $\tau \rightarrow 0$ and then set $\alpha \rightarrow \infty$, we have

$$\lim_{\alpha \rightarrow \infty} \lim_{\tau \rightarrow 0} \frac{\tau N^*(\tau, \alpha)}{T^*} \geq \frac{1}{\beta}.$$

Details of the proof can be found in the technical report [20].

6.2 The Min-weight Configuration Problem

We propose an algorithm for solving the min-weight problem with a guaranteed approximation ratio $\beta = 1.61$ by using previous results in the uncapacitated facility-location (UFL) problems [12].

The UFL problem can be stated as follows. Let \mathcal{F} be a set of facilities and let \mathcal{C} be a set of customers. Let $h_u > 0$ be the fixed cost for opening the facility $u \in \mathcal{F}$ and let $g_{uv} > 0$ be the service cost for each $u \in \mathcal{F}$ and $v \in \mathcal{C}$. We seek a subset of open facilities $\mathcal{F}_o \subseteq \mathcal{F}$, such that all customers are served and the sum of facility costs and service costs

$$\sum_{u \in \mathcal{F}_o} h_u + \sum_{v \in \mathcal{C}} \min_{u \in \mathcal{F}_o} g_{uv} \quad (19)$$

is minimized. A special type of UFL problems is called *metric uncapacitated facility-location* (metric-UFL) problems where the service costs are metric, that is,

$$g_{uv} + g_{u'v} + g_{u'v'} \geq g_{uv'}, \forall u, u' \in \mathcal{F}, \text{ and } v, v' \in \mathcal{C}. \quad (20)$$

In [12], Metric-UFL problems are solved by using an approximation algorithm with an approximation ratio 1.61.

We will convert the min-weight configuration problem (18) into an UFL problem with the mappings shown in Table 2:

(i) We set $\mathcal{F} = V_b$ and $\mathcal{C} = V_r$. The open facility set we are looking for is the set of active BSs

$$\mathcal{F}_o = \{u : z_u = 1, u \in V_b\}. \quad (21)$$

(ii) We assign a length d_{ij} for each edge of the WSN,

$$d_{ij} = \begin{cases} \lambda_i c_{st} r + \lambda_j c_{sr} r, & i, j \in V_b \cup V_r, \\ \lambda_i c_{1t} r, & i \in V_b, j = R, \\ \infty, & \text{otherwise,} \end{cases} \quad (22)$$

where λ_i is the i -th element of the weight vector λ .

(iii) For each sensor node $v \in V_r$ and each BS $u \in V_b$, we calculate the shortest path from sensor node v to the remote server S via BS u , which we denote by L_{uv} . We let the service cost g_{uv} for BS u forward the data packets from sensor node $v \in V_r$ be

$$g_{uv} = L_{uv} + \lambda_v c_c, \quad (23)$$

and we let the facility cost h_u for activating BS $u \in V_b$ be

$$h_u = \lambda_u c_{1c}. \quad (24)$$

The UFL problem constructed from the mappings in Table 2 is not metric in general, because the length of a path under the distance assignment (22) is not symmetric: the length of the shortest path from sensor node v to BS u is not equal to the reverse path from BS u to sensor node v . In the following theorem, we show the conditions that ensure the UFL to be metric.

THEOREM 3. *Given the parameters $c_c, c_{st}, c_{sr}, c_{1c}, c_{1t}, r$, and a weight vector λ , we construct an instance of the UFL problem from the min-configuration problem (18) by using the mappings shown in Table 2. The constructed UFL problem*

- *has the same optimal objective value as that of the min-weight configuration problem (18), and*
- *is metric (20) under the condition that $c_{sr} \leq c_{st} \leq 2c_{1t} + c_{sr}$.*

Details of the proof can be found in the technical report [20].

We then use the 1.61-approximation algorithm proposed in [12] to solve the constructed UFL problem. For simplicity of discussion, we define the set of unconnected customers as \mathcal{C}_u . Let the two-tuple (i, \mathcal{C}) denote a star that is composed of a facility $i \in \mathcal{F}$ and a set of customers $\mathcal{D} \subseteq \mathcal{C}$ connected to facility i . The cost of the star (i, \mathcal{D}) is defined as the total cost divided by the number of unconnected customers in \mathcal{D} :

$$\text{cost}(i, \mathcal{D}) = \begin{cases} \frac{h_i + \sum_{j \in \mathcal{D}} q_{ij}}{|\mathcal{C}_u \cap \mathcal{D}|}, & i \notin \mathcal{F}_o, \\ \frac{\sum_{j \in \mathcal{D}} q_{ij}}{|\mathcal{C}_u \cap \mathcal{D}|}, & i \in \mathcal{F}_o, \end{cases} \quad (25)$$

where

$$q_{ij} = \begin{cases} g_{ij}, & j \in \mathcal{C}_u, \\ \min(0, g_{ij} - \min_{k \in \mathcal{F}_o} g_{kj}), & j \notin \mathcal{C}_u. \end{cases}$$

The procedure of the algorithm is as follows:

- (i) At the beginning, we set $\mathcal{F}_o = \emptyset$ and $\mathcal{C}_u = \mathcal{C}$.
- (ii) As long as the set of unconnected customers $\mathcal{C}_u \neq \emptyset$, find the most cost-efficient star (i, \mathcal{D}) for all $i \in \mathcal{F}$ and $\mathcal{D} \subseteq \mathcal{C}$. Connect all customers in \mathcal{D} to i and set $\mathcal{C}_u = \mathcal{C}_u \setminus \mathcal{D}$. If $BS\ i \notin \mathcal{F}_o$, activate BS i and $\mathcal{F}_o = \mathcal{F}_o \cup \{i\}$.

We note that in step (ii), although the number of stars is exponential with the number of customers, it is easy to find

Algorithm 2: Approximation algorithm for solving the min-weight configuration problem

Input: The parameters $c_c, c_{st}, c_{sr}, c_{lc}, c_{lt}, r$, and a weight vector λ .

Output: A configuration $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) \in \mathcal{L}$.

- 1 Assign the lengths of edges in the WSN as shown in (22).
 - 2 Set $\mathcal{F} \leftarrow V_b$ and set $\mathcal{C} \leftarrow V_r$.
 - 3 Calculate the shortest path L_{uv} from each sensor node $v \in \mathcal{C}$ to the remote server S via each BS $u \in \mathcal{F}$.
 - 4 Assign the service costs g_{uv} ($u \in \mathcal{F}, v \in \mathcal{C}$) and the facility costs h_u ($u \in \mathcal{F}$) as shown in (23) and (24), respectively.
 - 5 Set $\mathcal{F}_o \leftarrow \emptyset$ and set $\mathcal{C}_u \leftarrow \mathcal{C}$.
 - 6 **while** $\mathcal{C}_u \neq \emptyset$ **do**
 - 7 Find the most cost-effective star (i, \mathcal{D}) where $i \in \mathcal{F}$ and $\mathcal{D} \subseteq \mathcal{C}$ with the cost of the stars defined in (25).
 - 8 Connect all sensor nodes in \mathcal{D} to BS i along the shortest paths with edge lengths defined in (22). Set $\mathcal{C}_u \leftarrow \mathcal{C}_u \setminus \mathcal{D}$.
 - 9 Enable BS i to be active and set $\mathcal{F}_o \leftarrow \mathcal{F}_o \cup \{i\}$.
-

the most cost-effective one star. It is sufficient to consider stars (i, \mathcal{D}_k^i) for $i \in \mathcal{F}$ and $k \in \{1, \dots, |\mathcal{C}_u|\}$, where \mathcal{D}_k^i denotes the set containing all connected customers $j \in \mathcal{C} \setminus \mathcal{C}_u$ whose $q_{ij} < 0$ and containing k unconnected customers $j \in \mathcal{C}_u$ with the smallest positive q_{ij} . Clearly, other stars cannot be more cost-effective (More details can be found in [12]).

We summarize the whole procedure for finding the approximate min-weight configuration as in Algorithm 2.

THEOREM 4. *Algorithm 2 guarantees to find a configuration $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{c}) \in \mathcal{L}$, such that the weight $\lambda^\top \mathbf{c}$ is at most $\beta = 1.61$ times the optimal objective value of the min-weight configuration problem (18) under the condition that $c_{sr} \leq c_{st} \leq 2c_{lt} + c_{sr}$.*

In summary, the results in both Theorems 2 and 4 yield that our proposed algorithm achieves, under mild conditions, a lifetime at least 62% of the optimal one.

7. SIMULATIONS

In this section, we will show how we evaluate the proposed scheme by running extensive simulations.

We consider a $150 \text{ m} \times 150 \text{ m}$ sensing field where we randomly generate a connected WSN with $|V_r| = 40$ and $|V_b| = 5$ (The default network size is small so that problem (5) can be solved exactly). All sensor nodes generate data with a rate of $r = 1$ and BSs do not generate data. Data generated by sensor nodes are transmitted to active BSs using short-range communication whose per-hop transmitting range is 40 m. Time is partitioned into slots with the length of a time slot $\tau = 1$. Let the constants of energy consumption rates be $c_{sc} = 1$, $c_{st} = 1$, $c_{sr} = 1$, $c_{lc} = 10$ and $c_{lt} = 1$ (We select these parameters to reflect the real energy consumptions of BSs and sensor nodes. More details can be found in our recent paper [19]). Let the initially available energy of all BSs be $e_i = 5000, \forall i \in V_b$ and let that of all sensor nodes be $e_i = 3000, \forall i \in V_r$. Let the parameter $\alpha = 10000$ in (14).

Here, we simulate five schemes: (i) *OneFixed*, the traditional scheme that uses one always-active BS, (ii) *MultiFixed*

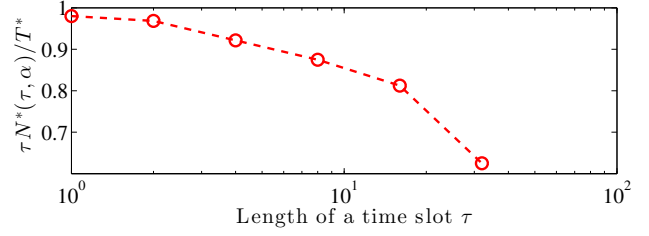


Figure 2: The ratio between the lifetime of MultiMove-A and that of MultiMove $\tau N^*(\tau, \alpha) / T^*$ versus the length of time slot τ . We see that the ratio decreases as τ increases. If τ is set small ($\tau < 10$), MultiMove-A yields a lifetime very close to the optimum achieved by MultiMove.

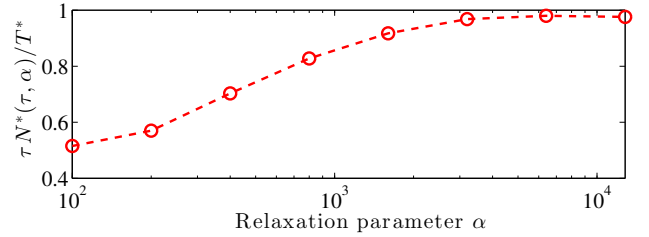


Figure 3: The ratio between the lifetime of MultiMove-A and that of MultiMove $\tau N^*(\tau, \alpha) / T^*$ versus the relaxation parameter α . We see that when α is sufficiently large ($\alpha > 1000$), MultiMove yields a lifetime very close to the optimum.

that uses multiple always-active BSs, where the locations of active BSs are optimized through (6), (iii) *OneMove* that schedules the virtual movement of one BS through (7), (iv) *MultiMove* that schedules the virtual movement of multiple BSs by solving (5) without approximation (considered as the optimum in the following), and (v) *MultiMove-A* that schedules the virtual movement of multiple BSs by using our proposed Algorithm 1 and Algorithm 2 to approximately solve (5). The details of MultiFixed, OneMove, MultiMove are discussed in Section 4.3. Without the fixed energy-consumption of long-range communication, both MultiFixed and MultiMove are optimal, they will always activate all BSs and use the maximum lifetime routing [7]. Because problem (5) is computationally hard, MultiMove is only applicable when the network is small.

7.1 Parameter Selection

As shown in Theorem 2, the performance of MultiMove-A depends on the parameters including the length of a time slot τ and the relaxation parameter α . We show how these parameters influence the ratio between the lifetime achieved by MultiMove-A and that by MultiMove $\tau N^*(\tau, \alpha) / T^*$.

In Figure 2, we show the ratio $\tau N^*(\tau, \alpha) / T^*$ versus the length of a time slot τ . When τ is small, MultiMove-A yields a lifetime very close to the optimum achieved by MultiMove, as the algorithm exploits the fine granularity of the time slot.

In Figure 3, we show the ratio $\tau N^*(\tau, \alpha) / T^*$ versus the relaxation parameter α in (14). When α is large, the relaxation of the objective function is precise, and MultiMove-A yields a lifetime very close to the optimum.

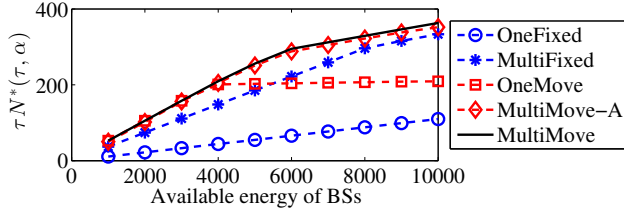


Figure 4: The network lifetime $\tau N^*(\tau, \alpha)$ versus the initially available energy of BSs when running different schemes..

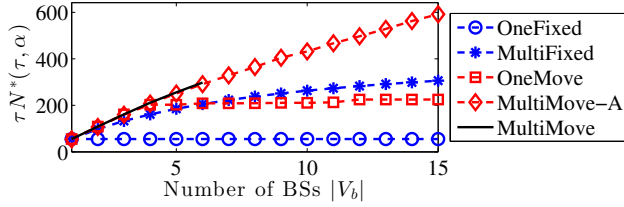


Figure 5: The network lifetime $\tau N^*(\tau, \alpha)$ versus the number of BSs $|V_b|$ when running different schemes..

7.2 Performance Comparison

To simulate different scenarios, we vary five parameters of the network: (i) the initially available energy of BSs, (ii) the number of BSs $|V_b|$, (iii) the number of sensor nodes $|V_r|$, (iv) the fixed energy-consumption rate of using long-range communication c_{lc} , and (v) the data rate of sensor nodes r .

We summarize the simulation results as follows: (i) In some scenarios, BSs are bottlenecks of the network lifetime, *e.g.*, when the initially available energy of BSs is low, when $|V_b|$ is small, when $|V_r|$ is large, when c_{lc} is large and when r is small. In these scenarios, OneMove outperforms MultiFixed, because OneMove improves the energy efficiency of BSs by rotating one active BS among multiple BSs. (ii) In other scenarios, sensor nodes are bottlenecks of the network lifetime, for example, when the initially available energy of BSs is high, when $|V_b|$ is large, when $|V_r|$ is small, when c_{lc} is small and when r is large. In these scenarios, MultiFixed outperforms OneMove, because MultiFixed reduces the energy consumption of sensor nodes by reducing the hop-distance from sensor nodes to BSs. (iii) In all scenarios, the lifetime achieved by MultiMove-A is always very close to the optimum achieved by MultiMove and is always longer than those achieved by both OneMove and MultiFixed.

Lifetime versus the initially available energy of BSs: In Figure 4, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the initially available energy of BSs. We fix the initially available energy of sensor nodes $e_i = 3000, \forall i \in V_r$, and we vary the initially available energy of BSs (all BSs have the same amount though). When the initially available energy of BSs is small, OneMove is very close to the optimum achieved by MultiMove and MultiMove-A. Therefore, in this scenario, virtually moving multiple BSs degrades into the scheme of virtually moving one BS as in [19]. When the initially available energy of BSs increases, the increase of lifetime of OneMove diminishes and Multi-Fixed outperforms OneMove, because the bottleneck of the network lifetime is shifted from BSs to sensor nodes.

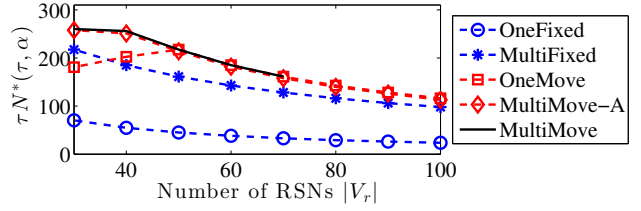


Figure 6: The network lifetime $\tau N^*(\tau, \alpha)$ versus the number of sensor nodes $|V_r|$ when running different schemes.

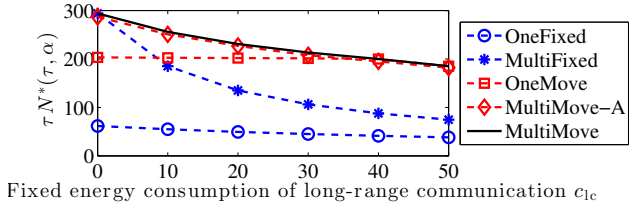


Figure 7: The network lifetime $\tau N^*(\tau, \alpha)$ versus the fixed energy-consumption of long-range communication c_{lc} when running different schemes.

Lifetime versus the number of BSs: In Figure 5, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the number of BSs $|V_b|$. We see similar results as that in Figure 4. When $|V_b|$ is small, OneMove outperforms MultiFixed and performs close to the optimum as MultiMove-A does. This is because when BSs are bottlenecks of the network lifetime, rotating one active BS is the optimal solution. When $|V_b|$ is large, sensor nodes become bottlenecks of the lifetime of the WSN, and MultiFixed outperforms OneMove.

Lifetime versus number of sensor nodes: In Figure 6, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the number of sensor nodes $|V_r|$. There is a clear trend that when $|V_r|$ increases, the lifetime of the network decreases. When $|V_r|$ is small, sensor nodes limit the lifetime of the WSN, and therefore OneMove performs poorly. When $|V_r|$ is large, BSs cause bottlenecks of the network lifetime, and OneMove performs close to the optimum achieved by MultiMove and MultiMove-A.

Lifetime versus the fixed energy-consumption of long-range communication: In Figure 7, we show the lifetime of the WSN $\tau N^*(\tau, \alpha)$ versus the fixed energy consumption rate of long-range communication c_{lc} . There is a trend that when c_{lc} increases, the lifetime of the network decreases. Still, MultiMove-A and MultiMove only decrease mildly because they can evenly distribute the high energy-consumption load among the whole WSN. Meanwhile, we see that the lifetime curve of OneMove is flat when c_{lc} is small, because the bottleneck of lifetime is due to the sensor nodes rather than BSs in these scenarios.

Lifetime versus the data rate of sensor nodes: In Figure 8, we show the achieved lifetime $\tau N^*(\tau, \alpha)$ versus the data rate r using different schemes. For better illustration, we set $e_i = 5000, \forall i \in V_b$ and $e_i = 1000, \forall i \in V_r$. We see similar results as that in Figure 7. When r is low, BSs are bottlenecks in the lifetime, and OneMove outperforms MultiFixed. When r becomes high, sensor nodes cause the bottleneck of the network lifetime, and the lifetime achieved by MultiFixed is very close to the optimum.

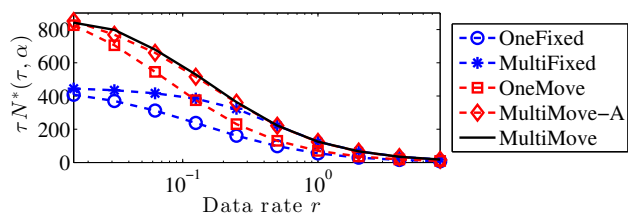


Figure 8: The network lifetime $\tau N^*(\tau, \alpha)$ versus the data rate of sensor nodes r when running different schemes.

8. CONCLUSION

In this paper, we have presented and evaluated the scheme of virtually moving the BSs in WSNs, where we adaptively re-elect an active subset of BSs. This scheme not only achieves a high energy-efficiency but also avoids the difficulty of physically moving the BSs. We have shown that the problem of virtually moving BSs is in fact NP-hard. We have proposed an adaptive algorithm for scheduling the virtual movement by using the constrained gradient method and using previous results in uncapacitated facility-location problems. Under mild conditions, this algorithm guarantees to yield a lifetime at least 62% of the optimum. The idea of load balancing by virtually moving devices can be used in other networked systems, including super-node selection in peer-to-peer networks and cooperative beamforming in cellular networks. As a consequence, the proposed scheduling algorithm has merits beyond the WSNs.

9. ACKNOWLEDGEMENTS

The work presented in this paper was supported (in part) by the Swiss National Science Foundation under grant number 200021-146423.

10. REFERENCES

- [1] G. Barrenetxea, B. Beresford-Lozano, and M. Vetterli. Lattice networks: capacity limits, optimal routing, and queueing behavior. *Networking, IEEE/ACM Transactions on*, 14(3):492–505, June 2006.
- [2] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wirel. Netw.*, 14(6):831–858, Dec. 2008.
- [3] L. Ben Saad and B. Tourancheau. Multiple mobile sinks positioning in wireless sensor networks for buildings. In *Sensor Technologies and Applications, 2009. SENSORCOMM.*, June 2009.
- [4] Y. Bi, L. Sun, J. Ma, N. Li, I. Khan, and C. Chen. Hums: An autonomous moving strategy for mobile sinks in data-gathering sensor networks. *Eurasip Journal on Wireless Communications and Networking*, 2007.
- [5] A. Bogdanov, E. Maneva, and S. Riesenfeld. Power-aware base station positioning for sensor networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [6] A. Z. Broder, A. M. Frieze, and E. Upfal. On the satisfiability and maximum satisfiability of random 3-cnf formulas. *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 322–330, 1993.
- [7] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(4):609–619, Aug 2004.
- [8] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.
- [9] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *IEEE Global Telecommunications Conference 2003*.
- [10] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. Comput.*, 37(2):630–652, May 2007.
- [11] P. Gonzalez-Brevis, J. Gondzio, Y. Fan, H. Poor, J. Thompson, I. Krikidis, and P.-J. Chung. Base station location optimization for minimal energy consumption in wireless networks. *IEEE Vehicular Technology Conference*, 2011.
- [12] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *J. ACM*, 50(6):795–824, Nov. 2003.
- [13] J. Luo and J. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. *2005. 24th Annual Joint Conference of the INFOCOM*, 2005.
- [14] J. Luo and J.-P. Hubaux. Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility. *IEEE/ACM Trans. Netw.*, 18(3):871–884, June 2010.
- [15] Y. Shi and Y. Hou. Some fundamental results on base station movement problem for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 2011.
- [16] Y. Tanaka, S. Imahori, M. Sasaki, and M. Yagiura. An lp-based heuristic algorithm for the node capacitated in-tree packing problem. *Computers and Operations Research*, 39(3):637–646, 2012.
- [17] Z. Vincze, R. Vida, and A. Vidacs. Deploying multiple sinks in multi-hop wireless sensor networks. In *Pervasive Services, IEEE International Conference on*, pages 55–63, July 2007.
- [18] Y. Yun and Y. Xia. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. *Mobile Computing, IEEE Transactions on*, 9(9):1308–1318, Sept 2010.
- [19] R. Zhang, F. Ingelrest, G. Barrenetxea, P. Thiran, and M. Vetterli. The beauty of the commons: Optimal load sharing by base station hopping in wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, PP(99):1–1, 2015.
- [20] R. Zhang, P. Thiran, and M. Vetterli. Virtually Moving Base Stations for Energy Efficiency in Wireless Sensor Networks. *Technical report*, <http://infoscience.epfl.ch/record/206994>.
- [21] G. Zussman and A. Segall. Energy efficient routing in ad hoc disaster recovery networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 682–691 vol.1, March 2003.