On Maximizing the Lifetime of Wireless Sensor Networks Using Virtual Backbone Scheduling

Yaxiong Zhao, Jie Wu, *Fellow, IEEE*, Feng Li, *Member, IEEE*, and Sanglu Lu, *Member, IEEE*

Abstract—Wireless Sensor Networks (WSNs) are key for various applications that involve long-term and low-cost monitoring and actuating. In these applications, sensor nodes use batteries as the sole energy source. Therefore, energy efficiency becomes critical. We observe that many WSN applications require redundant sensor nodes to achieve *fault tolerance* and *Quality of Service* (QoS) of the sensing. However, the same redundancy may not be necessary for multi-hop communication because of the light traffic load and the stable wireless links. In this paper, we present a novel sleep-scheduling technique called *Virtual Backbone Scheduling* (VBS). VBS is designed for WSNs has redundant sensor nodes. VBS forms multiple *overlapped* backbones which work alternatively to prolong the network lifetime. In VBS, traffic is only forwarded by backbone sensor nodes, and the rest of the sensor nodes turn off their radios to save energy. The rotation of multiple backbones makes sure that the energy consumption of all sensor nodes is balanced, which fully utilizes the energy and achieves a longer network lifetime compared to the existing techniques. The scheduling problem of VBS is formulated as the *Maximum Lifetime Backbone Scheduling* (MLBS) problem. Since the MLBS problem is NP-hard, we propose approximation algorithms based on the *Schedule Transition Graph* (STG) and *Virtual Scheduling Graph* (VSG). We also present an *Iterative Local Replacement* (ILR) scheme as a distributed implementation. Theoretical analyses and simulation studies verify that VBS is superior to the existing techniques.

Index Terms—Wireless sensor networks (WSNs), backbone scheduling, sleep scheduling, virtual backbone, energy-delay trade-off, connected dominating set, complexity analysis.

1 Introduction

The past two decades have witnessed the boom of *Wireless Sensor Networks* (WSNs), an enabling technology for various applications that involve long-term and low-cost monitoring, such as battlefield reconnaissance, building inspection, security surveillance, etc. In most WSNs, the battery is the sole energy source of the sensor node. Sensor nodes are expected to work on batteries for several months to a few years without replenishing. Thus, *energy efficiency* becomes a critical issue in WSNs.

Among the functional components of a sensor node, the radio consumes a major portion of the energy [1]. Various techniques are proposed to minimize its energy consumption. In this paper, we focus on *Backbone Scheduling* (BS), which dynamically turns off the radio of the sensor nodes to save energy. BS lets a fraction of the sensor nodes in a WSN turn on their radio to forward messages, which forms a backbone; the rest of the sensor nodes turn off their radio to save energy. This technique does not affect communication quality because those WSNs have *redundancy*. By redundancy, we mean that turning off the radio of some sensor nodes in a WSN does not affect the connectivity of the network. This redundancy results in more than necessary wireless links. Thus, it is possible to construct communication backbones to save energy. Specifically, we use *Connected Dominating Set* (CDS) algorithms to construct such backbones. A preliminary on CDS algorithms is given in Section 3 of the supplementary file.

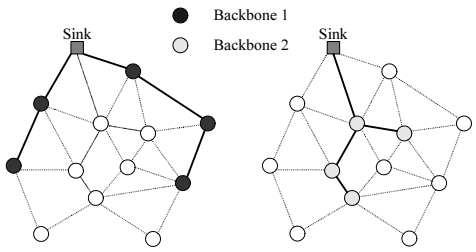


Fig. 1. An example of rotating two disjoint backbones in a (duty-cycled) WSN. The sink has an unconstraint energy supply and is implicitly included in all backbones.

However, a single backbone does not prolong the network lifetime. An intuitive idea is to construct multiple disjointed CDSs and let them work alternatively. This approach has been studied in [2] and is formulated as a *Connected Domatic Partition* (CDP) problem. Fig. 1 shows an example of two disjoint backbones.

In this paper, we propose *Virtual Backbone Scheduling* (VBS), a novel algorithm that enables fine-grained sleep-scheduling. VBS schedules multiple overlapped backbones so that the network energy consumption is evenly

- *Yaxiong Zhao and Jie Wu are with the Computer and Information Sciences Department of Temple University, E-mail: {yaxiong.zhao, jiewu}@temple.edu.*
- *Feng Li is with the School of Engineering and Technology, IUPUI: Indiana University-Purdue University Indianapolis, E-mail: fengli@iupui.edu.*
- *Sanglu Lu is with the State Key Laboratory for Novel Software Technology, Nanjing University, P. R. China, E-mail: sanglu@nju.edu.cn.*

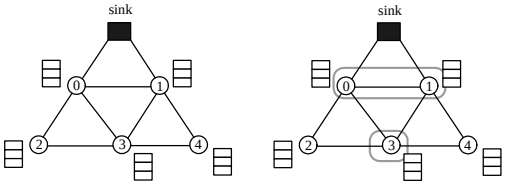


Fig. 2. A simple network consisting of five sensor nodes and a sink, where each sensor node has 3 units of energy. 1 unit of energy is consumed per unit of time. This graph only has one disjoint CDS formed by $\{sink, 0, 1\}$, $\{sink, 0, 3\}$, or $\{sink, 1, 3\}$. The network lifetime is 3 units of time using the CDP approach.

distributed among all sensor nodes. In this way, the energy of all of the sensor nodes in the network is fully utilized, which in turn prolongs the network lifetime. A motivating example is illustrated in Fig. 2. The figures show a network of five sensor nodes and one sink. The stack beside each node represents its initial energy. Assuming that all sensor nodes consume 1 unit of energy per unit of time, each sensor node can continuously work for 3 units of time. Since only one disjointed CDS, which is $\{sink, 0, 1\}$, $\{sink, 0, 3\}$, or $\{sink, 1, 3\}$, can be constructed, the network lifetime is 3 units of time. On the contrary, VBS schedules $\{sink, 0, 1\}$ to work for 1, $\{sink, 0, 3\}$ for 1, and $\{sink, 1, 3\}$ for 2 units of time, which achieves a network lifetime of 4 units of time. These backbones are overlapping (the sink is not considered overlapped because of its unconstrained energy supply). This example demonstrates that scheduling on a finer granularity can exploit the redundancy in the network and achieve a longer network lifetime than the CDP-based approach.

Nowadays, *Duty-Cycling* (DC) has become an integral technique for WSNs [3], [4], [5], [6], [7]. VBS combines BS with DC by letting backbone sensor nodes work in a duty-cycled fashion. Fig. 3 gives the schedules produced by VBS of the two backbones in Fig. 1.

In order to find the optimal schedule that maximizes the network lifetime by using VBS, we formulate the *Maximum Lifetime Backbone Scheduling* (MLBS) problem. We prove that it is NP-hard. We then present two centralized approximation algorithms to the MLBS problem. We also design a distributed implementation of VBS. We demonstrate, through extensive analyses and simulations, that our proposed solutions significantly prolong the network lifetime compared to the existing approach. Our contributions in this paper are as follows:

- We propose VBS, a combined backbone scheduling and duty-cycling method for WSNs with redundancy. VBS employs a fine-grained sleep-scheduling method, which significantly prolongs the network lifetime. We formulate the MLBS problem and prove its NP-hardness;
- We design two centralized approximation algorithms and a distributed implementation of VBS. Potential extensions are also discussed in the supplementary file;
- We conduct extensive theoretical analyses and simulation studies to verify the performance of VBS.

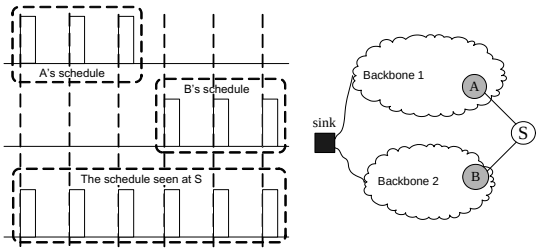


Fig. 3. Combining BS and DC to further prolong the network lifetime.

The rest of this paper is organized as follows. Related work is in Section 2. Section 3 presents the network model and defines the MLBS problem. Two centralized algorithms are presented in Section 4. Section 5 presents the distributed implementation of VBS. Simulation results are discussed in Section 6. The conclusion is in Section 7.

2 Related work

For a comprehensive discussion of the related work, please refer to the Section 2 of the supplementary file.

3 Network model and problem definition

In this section, we discuss the network model and the assumptions used in this paper. We then define the MLBS problem and prove its NP-hardness.

3.1 Model and Assumptions

We have the following assumptions about the WSNs that we consider in this paper. Sensor nodes are randomly placed in the field and are immobile thereafter. A battery is the sole energy source of the sensor nodes. There is only one sink in the network, which is always active and has an infinite power supply. All sensor nodes have an identical communication range (links are bidirectional). The power consumption of a sensor node is comprised of three parts: *sensing*, *computing*, and *radio*. For a typical sensor node, the radio is the most power-consuming part and may even dominate the energy consumption. Therefore, we only consider the scheduling of the radio.

Sensor nodes are duty-cycled and have the same working cycle. We define T continuous cycles as a *round*, where $T \geq 1$. T is a tunable parameter. At the beginning of each round, a backbone is selected to work in duty-cycling. Nodes that are not in the backbone will turn off their radios.

The *lifetime* of a sensor node is the time span from when it starts working to when its energy is depleted. The lifetime of a network is the minimum lifetime of all

of the sensors in the network. Because backbones rotate after each round, the lifetime is counted in rounds.

We also assume that the traffic load in the network is light. This assumption implies that the contention and the interference of the wireless channel is light too. Additionally, because we assume that sensor nodes are static, route failure is rare too. Actually, recent work [8], [9] show that the delivery ratio of a WSN in a real-world in-door environment can be as high as 99.98% in a continuous operation of four weeks. Based on these arguments, we will not consider the loss of the control packets in the design.

3.2 The Maximum Lifetime Backbone Scheduling Problem and its NP-hardness

In order to find the optimal schedule, we formulate the Maximum Lifetime Backbone Scheduling (MLBS) problem. Its definition is as follows:

A schedule in VBS is a set of backbones working sequentially in each $round$. Formally, we need to find a set of backbones, $B = \{B_1, B_2, \cdots, B_p\}$, and each backbone B_i works for T_i rounds. A schedule is, therefore, represented by a set of tuples, $\{\langle B_1, T_1\rangle, \cdots, \langle B_p, T_p\rangle\}$, that satisfy the following constraints:

- Connectivity: all $B_i \in B$ is a connected sub-graph of the network, and all other nodes are, at most, 1-hop away from a node in B_i. In other words, they are CDSs of the network.
- Energy constraints: the amount of energy consumed by any sensor node in the network at the end of the lifetime does not exceed its initial value.

The lifetime of a schedule is the lifetime of the network using this schedule to turn on and off the radio of the sensor nodes. The objective of the MLBS problem is to find that schedule that achieves the maximal network lifetime. Note that the backbones can be overlapped. The MLBS problem is NP-hard. Please refer to the Section 1 of the supplementary file for the detailed proof.

4 Centralized Approximation Algorithms for the MLBS Problem

Because the MLBS problem is NP-hard, we focus on designing approximation algorithms. In this section, we present two centralized approximation algorithms. CDS construction algorithms are used. A preliminary of the algorithms and a discussion of their path stretch problem are given in the Section 3 of the supplementary file.

4.1 A Scheduling Transition Graph-based Approximation Algorithm

Our first centralized approximation algorithm is based on a new concept called *Schedule Transition Graph* (STG). A STG is used to model a schedule in a WSN. Fig. 4 gives an example. As shown in the figure, the horizontal axis represents the time scale, counted in rounds. In each round, possible *states* are listed vertically, which are represented by ellipses. The number of possible states for each round is equal to the number of backbones. Each state contains a backbone and the corresponding *energy levels* (defined later). The state and the backbone have a one-to-one mapping. An initial state is placed at round 0 and is connected with all states in the first round to represent a starting point.

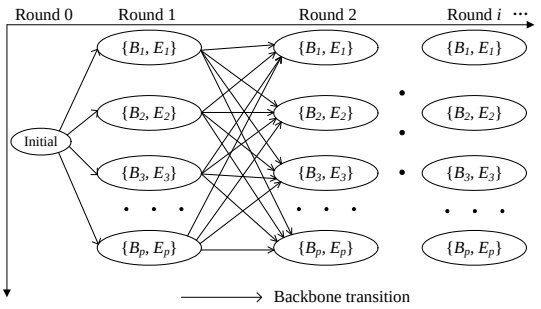


Fig. 4. The illustration of a STG. The initial state is attached as a common starting point for the scheduling.

Uni-directed *transition edges* connect states in one round to those in the next round. No backwards edges is allowed. Each edge represents the time elapse of 1 round. Since energy is used in each round, each edge also represents the consumption of energy. We assume that the sensor nodes in the backbone consume a fixed amount of energy in each round; all edges represent the same amount of energy consumption. The residual energy of all nodes is obtained by subtracting this value from the starting state of each transition edge. No transition is allowed if the energy of any sensor node of a state is depleted. It is clear that a directed path from the initial state corresponds to a schedule. Thus, the MLBS problem is thus to find the longest path in the STG.

4.1.1 Time span of an STG

The length of the horizontal direction of an STG is the maximum number of rounds that the network can run without depleting the energy of any sensor node, which is denoted as C. Given a network with a fixed topology and a finite amount of initial energy in each sensor node, the maximum round number is derived by dividing the sum of the initial energy of all nodes by the minimum amout of energy consumed in each round.

Firstly, we assume that each backbone node consumes a fixed amount of energy ϵ in each round. Because the MCDS is the lower-bound of the number of sensor nodes in a CDS of the WSN, the number of sensor nodes in any backbone is larger than that of the MCDS. Suppose that the size of the MCDS is n, then the minimum energy consumption in each round is at least $n \times \varepsilon$. Denote $\mathbb{E}$ as the initial energy of the sensor node in the network. Then, the total amount of energy that can be used is $|V|\mathbb{E}$, where $|V|$ is the number of sensor nodes in the network. The maximum round number C is given by Eq. 1:

$$C = \frac{|V|\mathbb{E}}{n \times \varepsilon} \tag{1}$$

Because n is in $O(|V|)$ and ε is a constant, C is in $O(\mathbb{E})$. Usually, the capacity of the batteries is limited, so we can treat $\mathbb{E}$ as a constant; C then becomes a constant too.

4.1.2 Energy level

The reason behind introducing the concept of *energy level* is to facilitate clean criteria for the search in the STG. We define the energy level σ of a WSN of $|V|$ sensor nodes as a tuple of all of the residual energy values of all of the sensor nodes in the network. Suppose that each sensor node V_i in a WSN has E_i^r units of residual energy, then the energy level of this network is $\langle E_1^r, E_2^r, \cdots, E_{|V|}^r \rangle$.

We further define the $\preceq$ $(less\ than)$ relation between two energy levels as follows. Two energy levels, σ_1 and σ_2, satisfy $\sigma_1 \preceq \sigma_2$, only if, for each $i \in \{1, 2, ..., |V|\}$ and $E_i^{r1} \in \sigma_1, E_i^{r2} \in \sigma_2$, there is $E_i^{r1} \leq E_i^{r2}$. $\sigma_1 < \sigma_2$ if $\sigma_1 \preceq \sigma_2$, and there is at least one i such that $E_i^{r1} < E_i^{r2}$.

An energy level is *zero* if at least one element is zero. Zero energy levels are less than any non-zero levels, and indicate the end of the network lifetime. The terminating state of any path in the STG contains a zero energy level. The energy level of the initial state of the STG is formed by the initial energy of all of the sensor nodes in the network.

4.1.3 Enumerating backbones

It is necessary to enumerate all possible backbones of the network in order to find the optimal solution. However, this is an exponential time operation, which is intractable. Instead, a polynomial number (in $|V|$) of backbones are constructed in our algorithm. In order to obtain better results, more backbones should be constructed. Since the sink has an unconstrained energy supply, we always add the sink as the first node into each candidate backbone. If two backbones, B_1 and B_2, satisfy $B_1 \subset B_2$, B_2 should not be included in the STG because any possible transition directed to B_2 will not yield a longer lifetime than those directed to B_1.

We use the CDS construction algorithms [10] iteratively. Initially, the priorities of all of the sensor nodes are randomly assigned. After each iteration, we reduce the priorities of the sensor nodes of the backbone constructed in this iteration, which lowers the probability that they are selected in the next iteration. We keep on constructing CDSs until the desired number of CDSs is obtained.

4.1.4 The STG-based algorithm

The approximation algorithm is based on dynamic programming. Its pseudo code is listed in Algo. 1. The search starts from the initial state. After a backbone transition, the state's energy levels are computed from those of the starting state of the transition. Each state keeps the larger energy levels. A path terminates when its associated energy level is zero. When all paths terminate, the longest path is found. The analysis of its complexity is given in Section 4.1 of the supplementary file.

Algorithm 1 STG-based algorithm

```
1: int CUR_ROUND ← 0;
2: repeat
3:   for each state S do
4:     Get the associated energy levels of S;
5:     Prune the resultant energy levels using the
       min() function;
6:     Select the energy level with the maximal mini-
       mum energy value.
7:     Set S's energy level to the the energy level with
       the maximum summation among the resultant
       energy levels;
8:   end for
9:   CUR_ROUND ← CUR_ROUND + 1;
10: until All the energy levels of the states in
    CUR_ROUND are zero;
11: Return the schedule represented by the path ending
    in CUR_ROUND
```

In searching for the longest path in the STG, we need to record the energy levels of each state. We define a function, $min()$, to prune *invalid* energy levels from a set of associated n energy levels, $S = \{\sigma_1, \sigma_2, ..., \sigma_n\}$, of each state:

$$min(S) = \{\sigma | \sigma \in S, \text{there is another state } \sigma' \in S \text{ such that } \sigma \preceq \sigma'\}$$

The $\preceq$ operator is defined in Section 4.1.2. The energy levels produced by this function are invalid because strictly lower energy levels cannot produce a longer lifetime. These energy levels are discarded. In order to reduce the complexity, we select the energy level that has the largest minimum value in the tuple as the associated energy level of each state. If there are still multiple energy levels associated with a state, the one that has the largest summation of all of the values of the energy level is kept. Lines 5, 6, and 7 of Algo. 1 execute these operations.

4.2 Virtual Scheduling Graph-based Approximation Algorithm

In this section, we aim to design a heuristic with less complexity than the STG-based algorithm. In STG, the energy and structure of the WSN are modeled separately. In this section, we propose a new concept called *Virtual Scheduling Graph* (VSG) that can model the energy and structure together, which facilitates an elegant greedy algorithm. In a VSG, a sensor node in the original network graph is converted into multiple $virtual\ nodes$, which are connected in such a way that their degrees represent the energy of the corresponding sensor node. A schedule can be obtained by applying any CDS construction algorithm on the VSG.

4.2.1 The definition of VSG

As stated before, each sensor node consumes a fixed amount of energy ε in each round when working as a

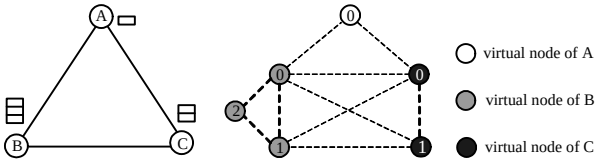


Fig. 5. The corresponding VSG (right) of a network of 3 sensor nodes (left). The virtual nodes of different ancestors are connected with an increasing index order. As a result, virtual node 2 of sensor node B is *isolated* because it has more energy and cannot be connected to the virtual nodes of A or C.

backbone node. We define a *virtual node* that corresponds to a sensor node as a node that contains ε energy. The original node is called the *ancestor*. An ancestor of E_r energy is divided into $\lceil \frac{E_r}{\varepsilon} \rceil$ virtual nodes. The virtual nodes of the same ancestor form a *virtual group*. Virtual nodes in the same virtual group are indexed. Two virtual groups are neighbors if their ancestors are neighbors in the original graph. The virtual nodes that have the same indexes are connected. A virtual node is *isolated* if it does not connect with any virtual node of other virtual groups. Fig. 5 shows an example of these concepts.

The VSG, $G_s(V', L')$, of a graph, $G(V, L)$, is constructed with the following *VSG rules* where V' and V are the sets of vertexes, and L' and L are the sets of edges:

- In a VSG, each ancestor is replaced by a clique of virtual nodes. The size of the clique corresponds to the energy of the ancestor.
- Nodes of two neighboring virtual groups are connected with an increasing index order until one group's virtual nodes are all connected. Suppose that two virtual groups have m and n virtual nodes, respectively, then only nodes of indexes in $[0, min\{m, n\} - 1]$ are connected.
- The *priority* of a virtual node is a tuple of its degree and ID, where the ID is its ancestor's ID plus the virtual node index (e.g., A0, A1, A2).

For example, nodes B and C in Fig. 5 have 3 and 2 units of energy, and they are replaced by 3 and 2 virtual nodes, respectively. Virtual nodes 0 and 1 of B and C are connected, but virtual node 2 of B is isolated.

We can view a VSG as a collection of scattered snapshots of the original graph in each round. A VSG preserves the connectivity of the original graph. The reason to use this approach is that it will be biased towards nodes with more energy. Nodes with more energy will cause "isolated" virtual nodes. In this way, the CDS construction algorithm is forced to pick virtual nodes with more energy (details are given in Section 4.2.3).

4.2.2 Pseudo minimum connected dominating set

We first define the *Pseudo Minimum Connected Dominating Set* (PMCDS) of the VSG. Virtual nodes of the same ancestor cannot be in the same PMCDS. If so, a node has to "borrow" energy from the "future". We show in lemma 1 that we are able to find a PMCDS in polynomial time using any MCDS approximation algorithm.

Definition 1 (Pseudo Minimum Connected Dominating Set): A Pseudo Minimum Connected Dominating Set of a VSG is a CDS that does not contain multiple virtual nodes of the same ancestor.

Lemma 1: PMCDS can be computed in polynomial time by using any MCDS approximation algorithm.

Proof: Given any CDS computed by a MCDS approximation algorithm, assuming that there exist multiple virtual nodes of a virtual group, which connect with the virtual nodes of other virtual groups, then these duplicate virtual nodes can be removed without affecting the connectivity and coverage. If a duplicate virtual node is isolated, there must be virtual nodes of its virtual group in the CDS that connect with other virtual groups so that it is safe to remove the isolated virtual nodes. Therefore, we are able to remove all duplicated virtual nodes. □

Theorem 1 below states that a PMCDS of the VSG corresponds to a CDS of the original graph.

Theorem 1: Given a graph $G(V, E)$ and its corresponding VSG $G_s(V', E')$, suppose that $\bigcup V_i'$ is a PMCDS of G', and $\bigcup V_i$ is the corresponding ancestor of $\bigcup V_i'$ in G, then $\bigcup V_i$ is a CDS of G.

Proof: The connectivity of the PMCDS $\bigcup V_i'$ directly corresponds to the original graph. At least one virtual node of an ancestor is in a PMCDS, so $\bigcup V_i$ must cover all nodes and is connected, which makes $\bigcup V_i$ a CDS. □

4.2.3 The VSG-based algorithm

We transform the MLBS problem in a WSN into a MCDS problem in its VSG. The algorithm is shown in Algo. 2. The algorithm uses the *marking process* (MP) for constructing the CDS. More details of the MP are in the supplementary file of this paper. Algo. 2 iteratively constructs PMCDSs of the VSG. Backbone nodes are removed from the VSG in each iteration. The VSG rules are applied after each iteration to preserve the correspondence between the original graph and the modified VSG so that Theorem 1 is still valid.

When all of the virtual nodes of any ancestor are removed, i.e., the energy of its ancestor is depleted, the algorithm ends. Like the STG-based scheduling, the sink is added into each backbone. A nice property of VSG is that nodes with more energy tend to have isolated virtual nodes and thus are forced to be selected in the MCDS algorithm. As shown in Fig. 5, the virtual node 2 of B is not connected with any virtual nodes of A or C. Thus, B is forced to be added into the CDS. The analysis of its complexity is given in Section 4.2 of the supplementary file.

5 The Distributed Implementation of VBS

In this section, we present a distributed implementation of VBS called *Iterative Local Replacement* (ILR). ILR lets

Algorithm 2 VSG-based algorithm

1: $S \quad \{\}$;
2: Construct the VSG $G_s(V', E')$ of $G(V, E)$;
3: **repeat**
4: Apply the *marking process* on $G_s(V', E')$;
5: Apply Rules 1&2 or Rule K on the induced graph;
6: Construct the PMCDS C' from the resultant CDS C;
7: Remove the highest indexed virtual nodes of the ancestors whose virtual nodes is in C' from $G_s(V', E')$;
8: Find the corresponding CDS C_i of C' in G;
9: $S \quad S \cup \{\langle C_i, T_i \rangle\}$;
10: **until** Any ancestor's virtual nodes are all eliminated from $G_s(V', E')$;
11: **return** S.

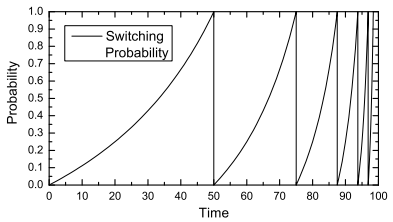


Fig. 6. The change of switching probability with time. The initial energy is 100 $joules$, and the consumption rate is 1 $joel/second$.

each backbone sensor node find replacement nodes to form a new CDS that preserves the connectivity of the network. Each sensor node of the backbone sensor only needs local information to do this. We summarize two issues that need to be considered in the design:

- Execution time: In each round, a backbone node that decides to switch its status collects or updates the information of its h-hop neighbors to find replacement nodes. The time used to perform these operations should be minimized.
- Quality of the results: Generally, more information (higher h) yields better results, which achieves a longer lifetime. However, it increases the message overhead and prolongs the execution time. The trade-off of these competing objectives should be carefully studied.

These two issues are contradictory. Various algorithms are available in literature, and they should be investigated carefully to choose the most appropriate one. We give two general optimizations that can be applied. Firstly, topology information does not need updating because sensor nodes are static. Secondly, energy consumption can be estimated according to the working statuses of sensor nodes. These two techniques avoid the costly message exchange of the ILR.

If all backbone nodes start the replacement simultaneously, many sensor nodes may contend the shared channel, which causes packet collision and loss. This situation may cause an increased execution time and more energy consumption. In order to avoid this problem, we employ a control-based scheme to make sure that sensor nodes do not perform replacement at the same time.

$$P_{switch} = \begin{cases} 1 - \frac{E_r}{\overline{E_r}} & \text{if } \overline{E_r} \geq E_T \wedge E_r \leq \overline{E_r} \\ 0 & \text{if } \overline{E_r} < E_T \vee E_r > \overline{E_r} \end{cases} \quad (2)$$

We assign a switching probability, P_{switch} (Eq. 2), to each backbone node. At the end of each round, backbone nodes switch statuses according to this probability. This probability is related to the residual energy of the backbone node and its h-hop neighbors. In Eq. 2, E_r is the residual energy of the backbone node, and $\overline{E_r}$ is the mean of the residual energy of its h-hop neighbors. Fig. 6 presents the change of switching probability with the time.

The rationale of Eq. 2 is that the probability rapidly increases with the time when the residual energy of the sensor nodes in the h-hop scope is lower. Backbone nodes with lower energy supplies are more "eager" to switch statuses, which helps balance the energy consumption of sensor nodes. There is a threshold E_T to stop the replacement when the residual energy is low; replacement becomes too expensive when there is not much energy left in the sensor nodes. This is given in the second equation in Eq. 2. The mean of the residual energy of h-hop neighbors is obtained from the information collected from neighbors up to h-hop away, as shown in the pseudocode of Algo. 3. The parameter h trades overhead for efficiency. The larger h is, the better the obtained results can be. When the residual energy of the backbone node is greater than the mean of its neighbors, the switching probability is also set to 0.

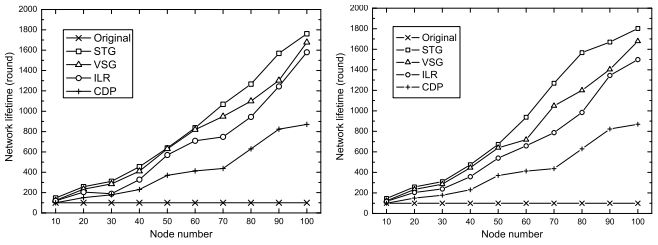
The pseudocode is listed in Algo. 3. ILR is executed by each backbone node at the end of each round. A backbone node that decides to switch broadcasts a message to "hold" h-hop neighbors, which keeps them awake for a longer time in order to complete the replacement. It then notifies its replacement nodes after the calculation is completed. The backbone node uses distributed algorithms, Rules 1&2 or Rule K, to find its replacement sensor nodes. The replacement sensor nodes found by the backbone node will be notified and will start working as backbone nodes in the next round. Both the message and time complexity of each replacement process are $O(\triangle^h)$, where $\triangle$ is the maximum node degree.

6 Performance Evaluation

We use simulations to evaluate the performance of VBS. The proposed algorithms are implemented in a customized simulator [11]. The simulator implemented the CDS construction algorithms that are used in this paper and has been used in previous work [10], [12]. We present the results of the network lifetime and the energy balance. The simulation results of the message delivery delay and the microscopic behaviors of ILR are in Section 6 of the supplementary file.

Algorithm 3 Iterative local replacement

loop
  At the beginning of each round;
  Sensor node N computes the switching probability P_{switch} using Eq. 2;
  if Decide to switch **then**
    Collect or update the h-hop information of N;
    Apply the *marking process* on the subgraph;
    Apply Rules 1&2 or Rule K on the induced graph using the residual energy as the priority;
    R The IDs of sensor nodes have more residual energy and can form a new CDS by replacing N;
    Notify each sensor node $N' \in R$;
  end if
end loop



(a) Rules 1&2 (b) Rule K

Fig. 7. Lifetime of networks with identical initial energy using Rules 1&2 and Rule K. The MP is applied before the pruning rules.

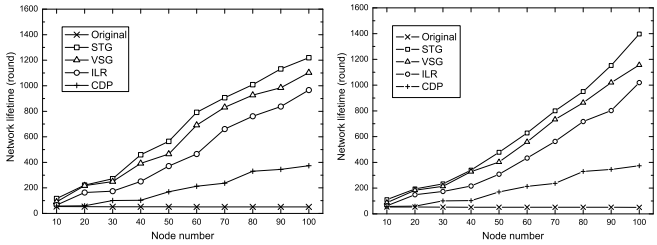
The networks are modeled as unit disk graphs [13]. Sensor modes are randomly placed in a square area. The sink is placed at the center of the area. All sensor nodes have the same transmission range. The number of sensor nodes is varied to model different network densities and scales. We assume that the sensor nodes in the backbone consume 1 unit of energy per round.

We compare VBS with the CDP-based method proposed in [2]. We use Rules 1&2 and Rule K to construct backbones. The exhaustive search for the optimal network lifetime is too time-consuming, even for small networks of 10 to 20 nodes; therefore, the optimal values are not presented. All results are obtained by averaging the results of 100 runs in random graphs with the same settings.

6.1 Network Lifetime

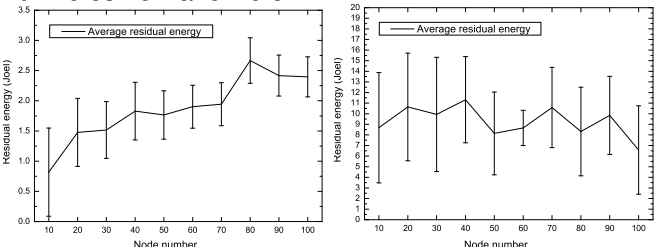
In this section, we present the results of the network lifetime achieved by our proposed algorithms. Two configurations are used: identical initial energy and imbalanced initial energy. Sensor nodes are deployed in a 500×500 area. The transmission range is fixed to 250 so that all of the networks generated are fully connected. The number of nodes in the network ranges from 10 to 100 with a step of 10. Since the area of the network is fixed, these settings vary the density of the sensor nodes.

Fig. 7 gives the achieved network lifetime of different algorithms. All sensor nodes have 100 units of initial



(a) Rules 1&2 (b) Rule K

Fig. 8. Lifetime of networks with uniformly-distributed initial energy in the interval $[50J, 100J]$, using MP together with Rules 1&2 and Rule K.



(a) Identical (b) Uniform distribution

Fig. 9. The mean and 90% confidential interval of the residual energy of all of the sensor nodes in a network of 100 nodes at the end of its network lifetime.

energy. The line labeled "original" represents the results of no sleep-scheduling. Rules 1&2 and Rule K are used to construct backbones. The STG-based algorithm produces the best results. The inferior performance of ILR is because it uses only local information. The backbone that is initially constructed using ILR may be of poor quality, e.g., it is too large or contains nodes of low energy. Another reason is that the transition between backbones is restricted to be local, so sensor nodes may not be replaced evenly. We also notice that Rule K achieves slightly longer lifetimes. This is because it constructs smaller-sized backbones, and the resultant backbones are more likely to be disjoint.

Fig. 8 presents the results in networks with uniformly-distributed initial energy. Each sensor node is assigned an initial energy drawn uniformly from $[50, 100]$. Because the lifetime is determined by the node with the minimum energy, the achieved lifetime when all nodes work is nearly halved, as shown in the line labeled "original". The lifetimes of all schemes in the assessment decrease drastically. However, our proposed schemes still achieve much longer lifetimes. The lifetime increases with network density because CDSs in denser networks are smaller and tend to be disjoint.

6.2 Energy Balance

We run three algorithms (STG, VSG, and ILR) once for a network of 100 sensor nodes, and then we record the residual energy of all sensor nodes at the end of the lifetime. Fig. 9 shows the means and 90% confidential intervals of the residual energy running the STG-based algorithm. The networks of the left figure are of an identical initial energy of 100 units. The networks of the

right figure have a uniformly-distributed initial energy in $[50, 100]$. The two figures show that the residual energy of all of the sensor nodes is small. The narrow confidential intervals indicate that the energy consumption is balanced. We can see that nodes in the network of imbalanced energy distribution have larger means and confidential intervals. VBS performs worse in networks with imblanced energy distribution, which is inevitable. The results of the VSG-based and ILR-based algorithms are similar to that of the STG-based one.

7 Conclusion

WSNs require energy-efficient communication to be able to work for a long period of time without human intervention. In this paper, we present a combined backbone-scheduling and duty-cycling method called VBS. VBS improves upon state-of-the-art techniques by taking advantage of the redundancy in WSNs. We formulate the MLBS problem to find the optimal schedule and prove its NP-hardness. Two centralized approximation algorithms with different complexities and performances are presented. Additionally, we design ILR, an efficient distributed implementation of VBS. We also conduct extensive theoretical analyses and simulation studies to verify the performance of VBS. In the supplementary file, related work, detailed proofs, and additional simulation results are given.

Acknowledgment

This research was supported in part by NSF grants ECCS 1128209, CNS 10655444, CCF 1028167, CNS 0948184, and CCF 0830289, and National Basic Research Program of China (973) No. 2009CB320705.

References

[1] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. of SenSys '04*. New York, NY, USA: ACM, 2004, pp. 188–200.

[2] C. Misra, R. Mandal, "Rotation of cds via connected domatic partition in ad hoc sensor networks," *IEEE Trans. on Mobile Computing*, pp. 488–499, 2009.

[3] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proc. of IEEE Infocom'02*, 2002, pp. 1567–1576.

[4] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proc. of ACM IPSN '05*, 2005, pp. 20–27.

[5] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *Proc. of ACM MobiHoc'06*, 2006, pp. 322–333.

[6] R. Cohen and B. Kapchits, "An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network," *IEEE/ACM Trans. on Netw.*, vol. 17, no. 2, pp. 570–581, 2009.

[7] Y. Li, W. Ye, and J. Heidemann, "Energy and latency control in low duty cycle mac protocols," in *Proc. of IEEE WCNC'05*, 2005, pp. 676–682.

[8] J. W. Hui and D. E. Culler, "IP is dead, long live IP for wireless sensor networks," in *Proc. of ACM SenSys'08*, 2008, pp. 15–28.

[9] L. Doherty, W. Lindsay, and J. Simon, "Channel-specific wireless sensor network path data," in *Proc. of ICCCN'07*, 2007, pp. 89–94.

[10] F. Dai and J. Wu, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," in *IEEE Trans. Parallel Distrib. Syst.* IEEE Press, 2004, pp. 908–920.

[11] "Wireless routing simulation suite," *http://sourceforge.net/projects/wrss/*.

[12] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless ad hoc and sensor networks," *J. Paral. Distrib. Comput.*, vol. 66, no. 7, pp. 947–958, 2006.

[13] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1-3, pp. 165–177, 1990.

Yaxiong Zhao received his B.E. degree in Transportation Engineering from Tongji University, Shanghai, China, in 2005; and M.E. degree in Software Engineering from Tsinghua University, Beijing, China, in 2008. He is currently a Ph.D. candidate in the Department of Computer and Information Sciences, Temple University, Philadelphia, PA. His research interests include various topics in the application and protocols of wireless networks. His current research focuses on the efficient communication between human-carried wireless devices and content-based networking.

Jie Wu Jie Wu (F'09) received the B.S. degree in computer engineering and the M.S. degree in computer science from Shanghai University of Science and Technology (now Shanghai University), Shanghai, China, in 1982 and 1985, respectively, and the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, in 1989. Jie Wu is an ACM Distinguished Speaker and an IEEE Fellow. He is a Chair and a Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA. Prior to joining Temple University, he was a Program Director with the National Science Foundation, Arlington, VA. He has published more than 550 papers in various journals and conference proceedings. His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu has served as an IEEE Computer Society Distinguished Visitor and is the Chairman of the IEEE Technical Committee on Distributed Processing (TCDP). He serves on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTERS and the Journal of Parallel and Distributed Computing. He is Program Co-Chair for IEEE INFOCOM 2011. He was also General Co-Chair for IEEE MASS 2006, IEEE IPDPS 2008, and DCOSS 2009.

Feng Li received his Ph.D. in Computer Science from Florida Atlantic University in Aug. 2009. His Ph.D. advisor is Prof. Jie Wu. He joined the Department of Computer, Information, and Leadership Technology at Indiana University-Purdue University Indianapolis (IUPUI) as an assistant professor in Aug. 2009. His research interests include the areas of wireless networks and mobile computing, security, and trust management. He has published more than 30 papers in conferences and journals.

Sanglu Lu Sanglu Lu received her B.S., M.S., and Ph.D. degrees from Nanjing University in 1992, 1995, and 1997, respectively, all in computer science. She is currently a professor in the Department of Computer Science & Technology and the State Key Laboratory for Novel Software Technology. Her research interests include distributed computing, wireless networks and pervasive computing.