Link Scheduling in Rechargeable Wireless Sensor Networks with a Dual-Battery System

Tony Tony[1], Sieteng Soh[2], Kwan-Wu Chin[3], and Mihai Lazarescu[4]

***Abstract*— This paper considers the problem of activating links in a rechargeable Wireless Sensor Network (rWSN). Unlike past works, it considers: (i) the energy harvesting time of nodes, (ii) a *battery cycle constraint* that accounts for *memory effects*, and (iii) nodes with a dual-battery system. It outlines a greedy algorithm that schedules links according to the earliest time in which a battery at the end nodes of each link can be discharged or is full. Our results show that equipping nodes with a dual-battery system decreases link schedules by up to 35.19% and 15.12% as compared to equipping nodes with a single battery with and without the said battery cycle, respectively. Such a system also respectively reduces the number of charge/discharge cycles by up to 15% and 87.13%. Finally, a longer energy harvesting time increases link schedules linearly, but has no impact on the number of charge/discharge cycles.**

***Index Terms*— Memory effects, channel access, varying energy harvesting time, link scheduling, charge/discharge.**

I. Introduction

Wireless Sensor Networks (WSNs) are of great interest to the Internet of Things (IoTs) community [1]. They have applications in the health industry, smart cities, and smart agriculture to name a few [2]. In these applications, sensor nodes collect and transmit data to a fusion center. Moreover, they are able to replenish their battery from environmental sources [3] to create so called rechargeable Wireless Sensor Networks (rWSNs). Hence, they are able to operate perpetually as long as their energy consumption rate is less than their energy harvesting rate.

A rWSN has a number of operational issues. First, the energy arrivals of nodes exhibit spatio-temporal properties. This means nodes have different energy harvesting rates. Second, as noted in [4], the battery of nodes suffers from *memory effects* if it is partially charged and discharged. This degrades a node's battery capacity. This degradation can be avoided by enforcing a so called *battery cycle* constraint, i.e., a node must charge (discharge) its battery completely before fully discharging (charging) its battery again [5], and also by equipping them with a dual-battery system [6].

The *third* issue is *link scheduling* [7], which determines when nodes activate their link(s) and thus governs the network capacity of a rWSN. To this end, this paper applies Time Division Multiple Access (TDMA), which ensures nodes do not experience collisions, which waste energy, and they only need to be active during their allocated time slot. To date, to ensure a high link/network capacity, a number of works such as [8] and [9] have seek to derive a short schedule by placing as many links as possible into each time slot.

Henceforth, we consider a novel *research aim*: derive a TDMA link schedule for a rWSN where nodes have heterogeneous energy harvesting rates and a dual-battery system. In a dual-battery system, when a battery is charged by an energy harvester, and at the same time, the other battery is discharged to power a node. A node switches the role of its batteries when one battery is fully charged and the other is fully discharged [6].

To explain the link scheduling problem, consider Fig. 1, where links (v_1, v_2), (v_1, v_3), (v_1, v_4), and (v_1, v_5) interfere with each other and hence cannot be activated concurrently. We use the Harvest-Store-Use (HSU) battery usage protocol [10], where any harvested energy must be stored in a battery before use. For example, node v_1 needs to wait for three time slots to accumulate one unit of energy, denoted by $v_1|3$, before the energy can be used no earlier than slot 4. Specifically, we will consider the following three scenarios. First, consider Fig. 1(a), where each node has one battery with a capacity of four units and no cycle constraint. Node v_1 needs to wait until v_2 can use one unit of energy at time $t = 7 + 1 = 8$ before it transmits a packet to v_2. Its second packet, which is destined to v_3, is transmitted at time $t = 8 + 1 = 9$; this is the earliest time v_3 has sufficient energy. Further, its third and fourth packet can be transmitted no earlier than time 12 and 15, respectively. Consequently, the schedule length is 15.

Next, consider Fig. 1(b) where each node has one battery with the same capacity as in Fig. 1(a). However, each battery has a battery cycle constraint. Thus, v_1 has to wait until slot $t = 3 \times 4 + 1 = 13$ to fully recharge and use its battery. However, at this time it cannot transmit because its neighbors' battery is yet to be fully recharged. That is, node v_2, v_3, v_4, and v_5 have to wait until slot $t = 7 \times 4 + 1 = 29$, $t = 33$, $t = 37$, and $t = 41$, respectively before their battery can be discharged. Thus, the example depicted in Fig. 1(b) has a longer schedule, i.e., 41, as compared to Fig. 1(a).

Lastly, consider Fig. 1(c), where each node has two batteries with the cycle constraint. The total energy available at nodes is the same as in Fig. 1(b). Node v_1 needs six slots to fully charge its first battery. Its second battery starts charging after the first battery is fully charged. This means at slot $t = 6 + 6 = 12$, the second battery is full and can be discharged at slot 13. Node v_1 needs to wait until slot $t = 6 + 1 = 7$ before its first battery can be used transmit a packet. However, it has to wait for its neighbors to fully charge their first battery. Note that nodes v_2, v_3, v_4, and v_5 require 15, 17, 19, and 21 slots, respectively before their first battery can be discharged. Now, node v_1 can transmit a packet to node v_2, v_3, v_4, and v_5 at slot $t = 15$, $t = 17$, $t = 19$ and $t = 21$, respectively. Thus, the schedule length is 21 versus 41 in Fig. 1(b). This example thus shows the benefit of employing a dual battery system at nodes. Note

that, at $t = 19$ and $t = 21$, node v_1 uses its second battery.

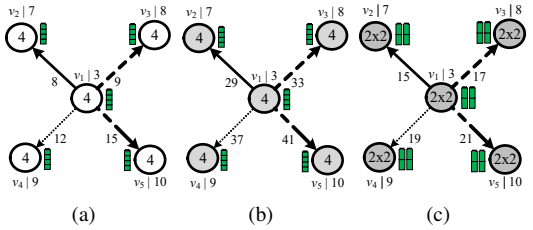


Fig. 1. An example for (a) single battery with no cycle constraint, (b) single battery with cycle constraint, (c) dual-battery with cycle constraint.

This paper contains the following *contributions*. First, we propose a TDMA link scheduler to maximize throughput in rWSNs where (i) sensor nodes have a different energy harvesting rate, (ii) each node is equipped with a dual-battery system, (iii) each battery has finite capacity, (iv) each battery has a *battery cycle* constraint, and (v) each link i has a weight $w_i \geq 1$ and must be scheduled at least w_i times. To the best of the authors' knowledge, no works have considered all of the above factors. Second, we develop a heuristic algorithm to generate TDMA link schedules, where it schedules links that can be activated at the earliest time when one of the dual-battery at their end nodes can be discharged to transmit/receive a packet. Next, we discuss prior works, and highlight their limitations.

II. Related Works

To the best of our knowledge, no works have considered our research aim. Sun *et al.* [8] proposed two link schedulers that consider links with and without a weight, and nodes with the HSU battery charging model. The authors [8] consider infinite battery capacity. Tony *et al.* [9] address the same problem as reference [8] and aim to minimize the schedule length but used the Harvest-Use-Store (HUS) [10] model, where nodes have a super capacitor, and they are able to use their harvested energy immediately or store it for later use. Similar to [8], the work in [9] applied batteries that are charged/discharged partially. Further, they [9] consider each battery has limited capacity and leaks over time, while reference [11] includes battery efficiency. The authors of [11] aim to minimize stored energy by scheduling links with the maximum weight first.

Recently, the authors of [12] proposed a link scheduler for rWSNs that addresses memory effects via a battery cycle constraint. In other words, their work [12] enforces a node to fully charge (discharge) its battery before it can be used (recharged). They used the HSU battery charging model and each node has varying energy harvesting rates. Note that the works reported in [8], [9], [11], and [12] consider nodes with a single rechargeable battery.

In summary, prior works assume nodes have a single battery. However, as highlighted in [5], this is unrealistic as in practice, if nodes charge and discharge their batteries simultaneously, their battery lifetime will degrade. Consequently, there is now interest in equipping nodes with a dual-battery system. To this end, we revisit the well-known link scheduling problem under a new practical setting.

III. Preliminaries

We model a rWSN as a directed graph $G(V, E)$, where each node $v_i \in V$ is a sensor node and each link $l_{i,j} \in E$ denotes a directed link from v_i to v_j. Each node v_i has a transmission range of $\mathcal{R}_i$. Let $||v_i - v_j||$ be the Euclidean distance between nodes v_i and v_j. A node v_i can transmit or receive packets to / from v_j if $||v_i - v_j|| \leq \mathcal{R}_i$. Each link $l_{i,j} \in E$ has weight $w_{i,j} \geq 1$, meaning the link must be activated $w_{i,j}$ times in the generated schedule. For instance, in Fig. 2(a), $w_{2,4} = 3$. Let ϵ (in Joule) be the energy consumed to transmit or receive one packet. We assume the energy usage for transmission and reception is equal.

We assume the protocol interference model [13], which considers (i) primary interference, where each node is half-duplex, and (ii) secondary interference, where a node, say A, while receiving a packet from its neighbor, say B, also receives a transmission from node C to D. The interference between links is modeled by a conflict graph $C_G(V', E')$ [14], which can be constructed for a graph $G(V, E)$ as follows: (i) each vertex in V' represents a link in E, i.e., $|V'| = |E|$, and (ii) each edge in E' represents two links of G that experience primary or secondary interference if they are active together. Fig. 2(b) shows the conflict graph C_G for the example rWSN depicted in Fig. 2(a).

A TDMA superframe or a link schedule consists of equal sized time slots. All links in each slot do not experience primary and secondary interference. Let $\mathcal{S}$ represent the superframe and $|\mathcal{S}|$ denote its length (in slots). Each slot is either *empty* or contains one or more non-interfering, concurrently active links. A slot is empty when all sensor nodes experience energy outage.

Nodes use the HSU [10] battery usage protocol. A node v_i is equipped with a harvester and two rechargeable batteries. Let $r_i \geq 1$ (in slots) be the *harvesting time* or the total number of slots that is required by a node v_i to accumulate 1ϵ of energy. Thus, the harvesting rate of a node is $\frac{\epsilon}{r_i}$ per time slot. Further, we denote the dual-batteries of node v_i as B_i^z, for $z \in \{1, 2\}$. For each pair of batteries (B_i^1, B_i^2) at node v_i, we call the first (second) battery as the *active (reserve)* battery. Let b_i^z (in unit ϵ) be the capacity of battery $z \in \{1, 2\}$ at node v_i. We assume both batteries have equal capacity, i.e., $b_i^1 = b_i^2$. Further, the capacity of each battery is sufficient to transmit or receive one packet, i.e., $b_i^z \geq 1\epsilon$.

Each battery follows the *battery cycle* constraint. This constraint requires each battery of node v_i to be (i) charged to its maximum capacity $b_{i,max}$ before it can be used, and (ii) discharged to its minimum capacity $b_{i,min}$ before it can be recharged. Thus, each battery's *usable* energy is $\tilde{b}_i^z = b_{i,max} - b_{i,min}$. Further, each battery can be in one of three modes: (i) *charging (C)*, (ii) *discharging (D)*, or (iii) *idle (I)*, i.e., when the battery is neither being charged or discharged. Without loss of generality, we assume each battery has an initial energy level of $b_{i,min}$, where $b_{i,min} < b_{i,max}$. Further, we assume $b_{i,min}$ and $b_{i,max}$ are integer values.

We use $\tilde{t}_i^{z+}$ and $\tilde{t}_i^{z-}$ to denote respectively the start and end charging time of battery z at node v_i. Similarly, t_i^{z+} and t_i^{z-} denote the start and end discharging time of battery z at node v_i, respectively. Following the HSU model, we have

$t_i^{z+} = \tilde{t}_i^{z-} + 1$ and $\tilde{t}_i^{z+} = t_i^{z-} + 1$. However, only one of the batteries at node v_i can be charged or discharged at any time t. Battery B_i^2 will start charging when B_i^1 stops charging. Hence, the start charging time of B_i^2 is equal to the end charging time of B_i^1, i.e., $\tilde{t}_i^{2+} = \tilde{t}_i^{1-}$. The active battery at node v_i becomes the reserve battery, and vice versa, when the energy level of battery B_i^1 reaches $b_{i,min}$ after packet transmission/reception and that of B_i^2 is fully charged. We assume the time to switch the role of batteries is negligible.

Let $\tilde{\tau}_i^z$ and τ_i^z respectively be the *charging time interval* and *discharging time interval* of battery z at node v_i, which are computed as $\tilde{\tau}_i^z = \tilde{t}_i^{z-} - \tilde{t}_i^{z+}$ and $\tau_i^z = t_i^{z-} - t_i^{z+}$. In other words, B_i^z is being charged during time interval $\tilde{\tau}_i^z$ and being discharged during time interval τ_i^z. The value of $\tilde{\tau}_i^z$ is dependent on r_i and $b_{i,max}$, while the length of τ_i^z is affected by $b_{i,min}$ and the number of times the batteries are used at the cycle. We note that the time interval $\tilde{\tau}_i^z$ has equal length in any cycle because r_i and $b_{i,max}$ are constant over all cycles. In contrast, the value of τ_i^z may vary at different cycles due to different energy usage between cycles.

Let $\tilde{b}_{i,t}^z$ and $b_{i,t}^z$ (in unit ϵ) be the amount of energy stored in battery z at node v_i at the start of slot t for a charging and discharging cycle, respectively. Thus, we have $\tilde{b}_{i,\tilde{t}_i^{z-}}^z = b_{i,max}$ and $b_{i,t_i^{z-}}^z = b_{i,min}$. The battery level of node v_i at the beginning of each *charging* cycle is given as $\tilde{b}_{i,\tilde{t}_i^{z+}}^z = b_{i,min}$. For each *discharging* cycle, it is $b_{i,t_i^{z+}}^z = b_{i,max}$.

Let T_i be the earliest time slot node v_i when it can transmit/receive a packet, i.e., when B_i^1 can be used to power v_i. The earliest time in which link $l_{i,j}$ can be scheduled is $t_{i,j} = \mathbf{max}(T_i, T_j)$. For each node v_i, we initialize $T_i = \tilde{\tau}_i^1 + 1$. Battery B_i^1 is initially in idle mode and it takes $\tilde{\tau}_i^1$ slots to reach level $b_{i,max}$. Following the HSU protocol, the energy can be used only one slot later. It is updated when the battery is discharged to transmit/receive a packet.

As an example, Fig. 2(a) shows the value of $b_{i,min}$, $b_{i,max}$, b_i^z, and r_i for each node v_i as well as the weight of each link $l_{i,j}$. One can compute the charging time interval for each battery to obtain $\tilde{\tau}_1^z = 4$, $\tilde{\tau}_2^z = 18$, $\tilde{\tau}_3^z = 14$ and $\tilde{\tau}_4^z = 12$; see Section IV for details. Thus, we have $T_1 = \tilde{\tau}_1^1 + 1 = 5$, $T_2 = 19$, $T_3 = 15$, and $T_4 = 13$. Therefore, the earliest time each link $l_{i,j}$ can be scheduled is computed as $t_{1,2} = \mathbf{max}(5, 19) = 19$, $t_{2,4} = 19$, $t_{4,3} = 15$.

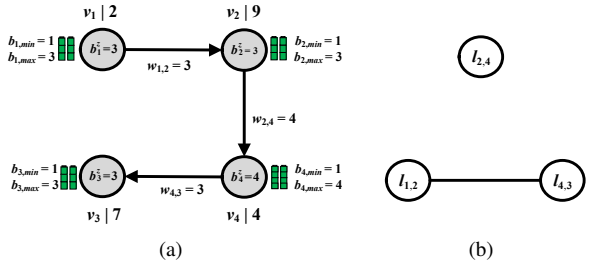
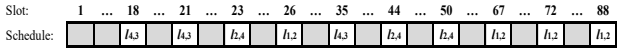


Fig. 2. A rWSN model. (a) Graph G, and its (b) conflict graph C_G.

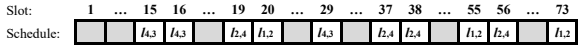
A. Problem Statement

Our problem is to generate the TDMA link schedule $\mathcal{S}$ with the minimum length $|\mathcal{S}|$ for a rWSN such that: (i) each node v_i is equipped with a dual-batteries system, (ii) each battery at node $v_i \in V$ satisfies the *battery cycle* constraint, (iii) each link $l_{ij} \in E$ can be scheduled at time slot t only if a battery of its end nodes is in discharging mode, and (iv) each link $l_{i,j} \in E$ is scheduled at least $w_{i,j}$ times in $\mathcal{S}$.

To illustrate the effect of link scheduling on $|\mathcal{S}|$, consider the example in Fig. 2. Fig. 3(a) shows one *feasible* link schedule with 88 slots. A schedule is *feasible* if it satisfies constraints (i), (ii), (iii), and (iv). The figure shows only non empty slots, i.e., each empty slot is represented as "…". Fig. 3(b) shows a shorter *feasible* schedule of $|\mathcal{S}| = 73$ slots. Our problem aims to generate the shortest *feasible* schedule $\mathcal{S}$. Note that, link scheduling in general is known to be NP-hard [15]. Lastly, we remark that our problem is the general version of the link scheduling problem that assumes nodes have no energy constraint. Thus, it is still NP-hard.



(a) One feasible TDMA link schedule.



(b) The optimal TDMA link schedule.

Fig. 3. TDMA link schedules for the rWSN depicted in Fig. 2. Gray colored slots show no packet transmissions/receptions.

IV. Solution

We first describe all possible states of (B_i^1, B_i^2) for node v_i, and their state transitions. Then we present three propositions relied upon by our greedy algorithm.

A. Battery States and Transitions

There are seven possible battery states at each node v_i. Each state S_k, for $k = 1, 2, \ldots, 7$, takes into account: (i) four of six possible combinations of battery mode, i.e., (I, I), (C, I), (D, I), and (D, C); note that the combination (C, C) and (D, D) are not possible because each node cannot charge or discharge two batteries at the same time, and (ii) the energy level of each battery, i.e., $b_{i,t}^z = b_{i,min}$, $b_{i,t}^z = b_{i,max}$, $b_{i,t}^z > b_{i,min}$, and $b_{i,t}^z < b_{i,max}$. Formally, a state k of batteries (B_i^1, B_i^2) at node v_i is defined as S_k = ({mode, energy level}, {mode, energy level}). Table I shows the seven states, i.e., $S_1 = (\{I, b_{i,t}^1 = b_{i,min}\}, \{I, b_{i,t}^2 = b_{i,min}\})$, …, $S_7 = (\{I, b_{i,t}^1 = b_{i,max}\}, \{I, b_{i,t}^2 = b_{i,max}\})$.

B. Key Properties

Proposition 1 and 2 are concerned with a battery in charging mode, while Proposition 3 is for a battery in discharging mode.

Proposition 1: The energy level of each battery $z \in \{1, 2\}$, in *charging* mode at node v_i at time slot t, for $\tilde{t}_i^{z+} \le t \le \tilde{t}_i^{z-}$, is $\tilde{b}_{i,t}^z = \mathbf{min}(b_{i,max}, b_{i,min} + (t - \tilde{t}_i^{z+})/r_i)$.

Proof: The maximum amount of energy that can be harvested from $\tilde{t}_i^{z+}$ to t is $\frac{t - \tilde{t}_i^{z+}}{r_i}$. However, $\tilde{b}_{i,t}^z$ is bounded by the upper limit of battery capacity $b_{i,max}$, which implies $\tilde{b}_{i,t}^z \le b_{i,max}$. ∎

Proposition 2 computes the number of slots required by node v_i to charge a battery B_i^z from $b_{i,min}$ to $b_{i,max}$, for $z \in \{1, 2\}$.

TABLE I

STATE AND STATE TRANSITION OF BATTERIES AT EACH NODE v_i

Possible States		Possible Transitions	
State	(B_i^1, B_i^2)	Transition	Explanation
S_1		$S_1 \rightarrow S_2$	B_i^1 starts charging, B_i^2 is idle.
S_2		$S_2 \rightarrow S_2$	B_i^1 is being charged; it is not full.
		$S_2 \rightarrow S_3$	B_i^1 is being charged; it is full.
S_3		$S_3 \rightarrow S_4$	B_i^1 is used, B_i^2 starts charging.
		$S_3 \rightarrow S_6$	B_i^1 is not used, B_i^2 starts charging
S_4		$S_4 \rightarrow S_2$	B_i^1 is empty, B_i^2 is not full; swap B_i^1 and B_i^2.
		$S_4 \rightarrow S_3$	B_i^1 is empty, B_i^2 is full; swap B_i^1 and B_i^2.
		$S_4 \rightarrow S_4$	B_i^1 is not empty, B_i^2 is not full.
		$S_4 \rightarrow S_5$	B_i^1 is not empty, B_i^2 is full.
S_5		$S_5 \rightarrow S_3$	B_i^1 is empty; swap B_i^1 and B_i^2.
		$S_5 \rightarrow S_5$	B_i^1 is not empty.
S_6		$S_6 \rightarrow S_4$	B_i^1 is used, B_i^2 is not full.
		$S_6 \rightarrow S_5$	B_i^1 is used, B_i^2 is full.
		$S_6 \rightarrow S_6$	B_i^1 is not used, B_i^2 is not full.
		$S_6 \rightarrow S_7$	B_i^1 is not used, B_i^2 is full.
S_7		$S_7 \rightarrow S_5$	B_i^1 is used.
		$S_7 \rightarrow S_7$	B_i^1 is not used.

 : Idle ($b_{i,t}^z = b_{i,min}$) : Charging ($b_{i,t}^z > b_{i,min}$)

 : Idle ($b_{i,t}^z = b_{i,max}$) : Discharging ($b_{i,t}^z < b_{i,max}$)

Proposition 2: The charging time interval of battery B_i^z at node v_i is computed as $\tilde{\tau}_i^z = r_i(b_{i,max} - b_{i,min})$.

Proof: We set $t = \tilde{t}_i^{z-}$ in Proposition 1 to generate the maximum energy level of the battery, i.e., $b_{i,max}$. We have $b_{i,max} = b_{i,min} + \tilde{\tau}_i{}^z/r_i$. Thus, $\tilde{\tau}_i^z$ is as shown in Proposition 2. ■

Proposition 3 computes the next earliest time in which node v_i can transmit/receive another packet.

Proposition 3: Let $t_i \geq \tilde{\tau}_i^1 + 1$ be the time in which node v_i last transmit/receive a packet. The next earliest time slot before node v_i can use its battery again to transmit/receive a packet is $T_i = t_i + \sigma_{i,t_i} \times r_i \times (b_{i,max} - b_{i,t_i}^2) + 1$.

Proof: The next earliest time T_i depends on the remaining energy level of active battery B_i^1 and reserve battery B_i^2 at node v_i at time t_i, i.e., b_{i,t_i}^1 and b_{i,t_i}^2, respectively. We consider two cases: (i) $b_{i,t_i}^1 = b_{i,min}$, and (ii) $b_{i,t_i}^1 > b_{i,min}$. For case (i), B_i^1 needs to be recharged and node v_i checks the energy level of B_i^2 at time t_i, i.e., b_{i,t_i}^2. We consider two sub-cases: (i.a) $b_{i,t_i}^2 = b_{i,max}$, and (i.b) $b_{i,t_i}^2 < b_{i,max}$. In both sub-cases, we use Proposition 1 to compute b_{i,t_i}^2. For sub-case (i.a), node v_i switches the role of its batteries from reserve (active) to active (reserve) since B_i^2 is fully charged. Thus, the next earliest time node v_i can transmit/receive a packet is in the next slot, i.e., $T_i = t_i + 1$. Thus, σ_{i,t_i} is set to zero. For sub-case (i.b), node v_i waits for γ_i slots for B_i^2 to be fully charged before it switches the role of its batteries. The next earliest time node v_i can transmit/receive another packet includes (i) the last time the node v_i transmit/receive a packet, (ii) time duration γ_i to reach its maximum level $b_{i,max}$ from the battery's level at time t_i, i.e., $\gamma_i = r_i \times (b_{i,max} - b_{i,t_i}^2)$, and (iii) one slot delay before the stored energy in B_i^2 can be used as required by the HSU model. Hence, $T_i = t_i + r_i \times (b_{i,max} - b_{i,t_i}^2) + 1$. For this sub-case, σ_{i,t_i} is set to 1. For case (ii), B_i^1 at node v_i has sufficient energy to transmit/receive another packet at time t_i. However, as a node can transmit/receive one packet at a time, the next earliest time node v_i can transmit/receive another packet is in the next slot, i.e., $T_i = t_i + 1$. For this case, σ_{i,t_i} is set to 0. This completes the proof. ■

C. Algorithm

We are now ready to explain **Link Scheduling with Dual Battery System (LSDBS)**. It aims to schedule all non-interfering links at the earliest possible time slot. At time $t = 0$, B_i^1 and B_i^2 are waiting to be charged. Further, both have an initial energy level of $b_{i,min}$. Finally, the initial charging time of B_i^1, i.e., $\tilde{t}_i^{1+}$ is set to zero.

Lines 1-3 calls *INIT*(.) to perform the following: (i) Use Proposition 2 to compute the charging time interval for each battery at node v_i, i.e., $\tilde{\tau}_i^1$ and $\tilde{\tau}_i^2$; (ii) Set the start of charging time of B_i^2, i.e., $\tilde{t}_i^{2+} = \tilde{\tau}_i^1$; (iii) Set $T_i = \tilde{\tau}_i^1 + 1$; and (iv) Set the energy level of B_i^1 at time T_i, i.e., $b_{i,T_i}^1 = b_{i,max}$.

Lines 4-6 compute $t_{i,j}$, i.e., the earliest time link $l_{i,j}$ can be scheduled. *Line 7* creates a set K that stores links $l_{i,j}$ that have the earliest activation time. *Line 8* then uses function *ORDER*(K) to to store in set K' links in set K in order of decreasing weight $w_{i,j}$. Links with equal $w_{i,j}$ are sorted in decreasing node degree of its end nodes and for a tie, links are sorted in increasing order of their node labels. *Line 9* sets t with the earliest slot, i.e., **min**$\{t_{i,j}\}$. *Lines 10-29* repeatedly schedule each link $l_{i,j} \in K'$ in order. Each selected link in *Line 12* does not cause interference or is interfered by links that have been scheduled in slot t; see the condition in *Line 11*. Each slot in $\mathcal{S}$ is initially empty. Note that function *CONFLICT()* uses a matrix M of size $|E|^2$ that contains Boolean variables to represent a conflict graph; i.e., $M[a, b]$ is set to one if there is an interference between links a and b. *Line 13* decreases the weight of each selected link $l_{i,j}$ by one. Once the link weight is equal to zero, *Line 15* removes the link from contention; see *Lines 14-16*.

Line *17* sets the last time an active battery at the end nodes of selected link $l_{i,j}$ is used, i.e., t_i and t_j, to the current time t. Further, *Lines 18-19* compute the energy level of the active battery at node v_i and v_j at time t after being used. *Lines 20-27* compute the following for each node $x \in \{i, j\}$. *Line 21* uses function *COMP_*$T_\alpha(.)$ that implements Proposition 3 to update the next earliest time node x can transmit/receive another packet. *Line 22* uses function *COMP_*$b_\alpha^z(.)$ to compute the energy level of B_x^1 at time T_x, which depends on the remaining energy level at time t_x, i.e., b_{x,t_x}^1. More specifically, if b_{x,t_x}^1 is larger than $b_{x,min}$, then b_{x,T_x}^1 is set to b_{x,t_x}^1 since the active battery still can be used. Otherwise, b_{x,T_x}^1 is set to $b_{x,max}$. For the latter case, i.e., $b_{x,T_x}^1 = b_{x,max}$, *Line 24* switches the role of batteries at node v_x. Then *Line 25* computes the start charging time of B_x^2, i.e., $\tilde{t}_x^{2+}$. Finally, the steps from *Line 4* is repeated until all links are scheduled, i.e., $w_{i,j} = 0$.

For example, consider Fig. 2. In *Line 2*, *INIT*(.) obtains $\tilde{\tau}_1^z = 4$, $\tilde{\tau}_2^z = 18$, $\tilde{\tau}_3^z = 14$, and $\tilde{\tau}_4^z = 12$. It sets $\tilde{t}_1^{2+} = 4$, $\tilde{t}_2^{2+} = 18$, $\tilde{t}_3^{2+} = 14$, and $\tilde{t}_4^{2+} = 12$. Further, it computes $T_1 = 5$, $T_2 = 19$, $T_3 = 15$, and $T_4 = 13$. Finally, it initializes $b_{1,T_1}^1 = 3$, $b_{2,T_2}^1 = 3$, $b_{3,T_3}^1 = 3$, and $b_{4,T_4}^1 = 4$. *Lines 4-6* compute $t_{1,2} = 19$, $t_{2,4} = 19$, and $t_{4,3} = 15$. *Line 7* inserts

links $l_{4,3}$ into the set K. *Line 8* obtains $K' = \{l_{4,3}\}$, while *Line 9* sets $t = 15$. *Line 12* inserts $l_{4,3}$ into $\mathcal{S}[15]$. *Line 13* gets $w_{4,3} = 2$. *Line 17* sets $t_4 = t_3 = 15$. *Lines 18-19* compute $b^1_{4,15} = 3$ and $b^1_{3,15} = 2$. *Lines 20-27* obtain $T_4 = 16$, $b^1_{4,T_4} = 3$ and $T_3 = 16$, $b^1_{3,T_3} = 2$. *Line 30* repeats the steps from *Line 4* until all links are scheduled. Finally, LSDBS produces the link schedule $\mathcal{S}$ in Fig. 3(b).

Algorithm 1 LSDBS

Input: $G(V, E)$, r_i, b^z_i, $b_{i,max}$, $b_{i,min}$, $w_{i,j}$, and C_G
Output: Superframe $\mathcal{S}$

1: **for** each node $v_i \in V$ **do**
2: *INIT*$(\tilde{\tau}^1_i, \tilde{\tau}^2_i, \tilde{t}^{2+}_i, T_i, b^1_{i,T_i})$
3: **end for**
4: **for** each link $l_{i,j} \in E$ **do**
5: $t_{i,j}$ = **max**(T_i, T_j)
6: **end for**
7: K = {node $l_{i,j}$ in C_G with **min**$\{t_{i,j}\}$}
8: K' = *ORDER*(K)
9: $t \leftarrow$ **min**$\{t_{i,j}\}$
10: **for** each $l_{i,j} \in K'$ **do**
11: **if** NOT *CONFLICT*$(l_{i,j}, \mathcal{S}[t])$ **then**
12: $\mathcal{S}[t] \leftarrow \mathcal{S}[t] \cup l_{i,j}$
13: $w_{i,j} \leftarrow w_{i,j} - 1$
14: **if** $w_{i,j}$ = 0 **then**
15: remove node $l_{i,j}$ from C_G
16: **end if**
17: $t_i \leftarrow t_j \leftarrow t$
18: $b^1_{i,t_i} \leftarrow b^1_{i,T_i} - 1$
19: $b^1_{j,t_j} \leftarrow b^1_{j,T_j} - 1$
20: **for** each node $x \in \{i, j\}$ **do**
21: $T_x \leftarrow$ *COMP_*$T_\alpha(x)$
22: $b^1_{x,T_x} \leftarrow$ *COMP_*$b^z_\alpha(x)$
23: **if** $b^1_{x,T_x} = b_{x,max}$ **then**
24: *SWAP*(B^1_x, B^2_x)
25: $\tilde{t}^{2+}_x \leftarrow T_x - 1$
26: **end if**
27: **end for**
28: **end if**
29: **end for**
30: **repeat** Line 4-29 **until** all $w_{i,j}$ = 0

V. Evaluation

We have implemented LSDBS in C++ and conducted our experiments on a computer with an Intel Core i7-9700T CPU @2.00 GHz and 16GB of RAM. We consider arbitrary networks with 10 to 50 nodes randomly deployed on a 40×40 m^2 area. Each node has a transmit and interference range of 15 and 30 meters, respectively. Our results are an average over 100 random node deployments. The average number of links $|E|$ is 28, 125, 273, 470, and 758 for rWSNs with 10, 20, 30, 40, and 50 nodes, respectively.

A. Effect of Battery Cycle Constraint

To study the effect of battery cycle constraint on the superframe length $|\mathcal{S}|$, we compare LSDBS, LSBCC [12], and LSNBC. Briefly, LSBCC is a version of LSDBS in which each node is equipped with only one battery. On the other hand, LSNBC is a version of LSBCC without the battery cycle constraint. We consider 10 to 50 nodes with $r_i = 5$. The total usable energy of *two* batteries when nodes use LSDBS is set to be equal to the *single* battery in LSBCC and LSNBC. Specifically, each battery in LSDBS has $\tilde{b}^z_i = 5\epsilon$, i.e., it has $b_i = 5\epsilon$, $b_{i,min} = 0\epsilon$, and $b_{i,max} = 5\epsilon$. Thus, the total usable energy of each node v_i in LSDBS is $2 \times 5 = 10\epsilon$. On the other hand, the battery in LSBCC and LSNBC has capacity of $b_i = 10\epsilon$, $b_{i,min} = 0\epsilon$, and $b_{i,max} = 10\epsilon$, i.e., its usable energy is $\tilde{b}_i = 10\epsilon$. Each link weight is randomly fixed to a value between 1 and 5, i.e., $w_{i,j} = [1, 5]$.

Effect on Superframe Length: Fig. 4 shows that the $|\mathcal{S}|$ produced by LSDBS is shorter than LSBCC. In a rWSN with ten nodes, LSDBS produces 82 less slots (35.19% shorter) as compared to when using LSBCC. The results are consistent for other networks with $|V| = 20, 30, 40, 50$ nodes; LSDBS produces superframes that are 27.85%, 27.45%, 27.03%, and 27.16% shorter than in LSBCC, respectively. This is because nodes that use LSDBS only need to wait for one battery, that has half usable energy of the battery in LSBCC, to be charged to its maximum level before powering their load. Fig. 4 shows that schedules generated by LSBDS are also shorter than those in LSNBC. Specifically, for $|V| = 10$, 20, 30, 40, and 50, LSDBS produces $|\mathcal{S}|$ that are respectively $9.04\%, 13.41\%, 14.26\%, 14.8\%$, and 15.12% shorter than in LSNBC. As shown in Fig. 4, the difference between schedule length of LSNBC and LSDBS increases when $|V|$ increases from 10 to 50. The reason is because superframe length is affected by the number of slots required by each node to accumulate energy as well as the number of links it has to activate. Each node in LSNBC needs $r_i + 1 = 6$ slots before it can activate one link, while that in LSDBS requires $r_i \times \tilde{b}^z_i = 25$ slots to charge one battery to its maximum level that can be used to activate five links. Note that, on average, each node in network with $|V| = 10, 20, 30, 40$, and 50 has $7, 16, 20, 30$, and 38 links to activate, respectively. Thus, a node that needs to activate 20 (30) links in LSNBC requires at least $20 \times 6 = 120$ (180) slots, which is 15 (25) slots longer than the node in LSDBS that on average requires $25 \times 4 + 5 = 105$ (155) slots.

Effect on Charge/Discharge Cycles: As shown in Fig. 5, the total number of charge/discharge cycles in LSBCC is larger than that in LSDBS - an average from two batteries. For example, when $|V| = 10$, LSBCC has three cycles more than in LSDBS; i.e., 15% more. The results are consistent for networks with $20, 30, 40, 50$ nodes, Specifically, LSBCC needs 6.17%, 4.62%, 3.4%, and 2.77% more cycles than LSDBS, respectively. Let $\mathcal{E}_i$ be the amount of energy (in ϵ) used at node v_i to schedule all of its incident links. Thus, each battery of v_i in LSDBS (LSBCC) needs $\lceil \mathcal{E}_i / \tilde{b}^z_i \rceil$ ($\lceil \mathcal{E}_i / \tilde{b}_i \rceil$) charge/discharge cycles to harvest $\mathcal{E}_i$ of energy. Note that $\tilde{b}_i = 2 \times \tilde{b}^z_i$, and thus we have $\lceil \mathcal{E}_i / \tilde{b}^z_i \rceil / 2 \leq \lceil \mathcal{E}_i / \tilde{b}_i \rceil$), i.e., charge/discharge cycles of each battery in LSDBS is no more than that of the battery in LSBCC. Fig. 5 also shows that the number of charge/discharge cycles produced by LSDBS is less than LSNBC. More specifically, LSDBS requires $84.38\%, 85.61\%, 86.47\%, 86.92\%$, and 87.13% fewer cycles than LSNBC for $|V|$ = 10, 20, 30, 40, and 50 nodes, respectively. This is because each battery in LSDBS has to be charged only when its energy is equal to the minimum level. The remaining experiments only use LSDBS.

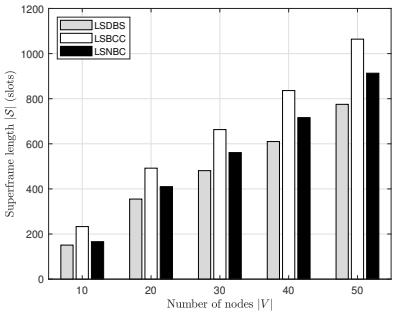


Fig. 4. LSDBS, LSBCC, LSNBC on $|\mathcal{S}|$.

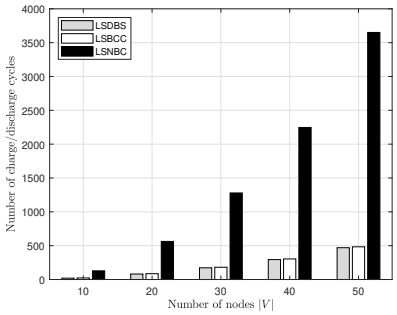


Fig. 5. LSDBS, LSBCC, LSNBC on cycles.

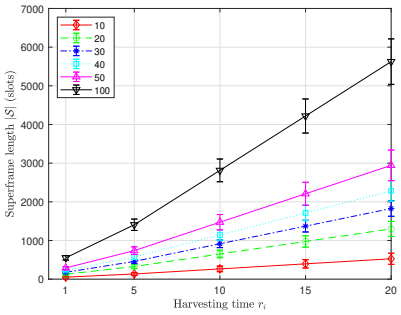


Fig. 6. Effect of harvesting time r_i on $|\mathcal{S}|$.

B. *Effect of Harvesting Time*

This section investigates the effect of r_i on $|\mathcal{S}|$ and charge/discharge cycles. In this simulation, we consider various r_i values, namely 1, 5, 10, 15, 20 slots in a rWSN with 10 to 50 nodes. We set $b_i^z = 3\epsilon$, $b_{i,min} = 1\epsilon$, $b_{i,max} = 3\epsilon$, and each link has weight $w_{i,j} = 3$.

Effect on Superframe Length: Fig. 6 shows that increasing r_i produces a longer superframe. Specifically, for 10 nodes, when r_i is increased from one to five, $|\mathcal{S}|$ jumps from 52 to 134 slots - an increase of 1.58 times. Similarly, when r_i increases from 5 to 20, i.e., to 10, 15, and 20, $|\mathcal{S}|$ increases by 0.98, 0.49, and 0.33 times, respectively. We observe similar trends for 20, 30, 40, and 50 nodes. For example, for 50 (100) nodes, when r_i increases from 1 to 20, $|\mathcal{S}|$ increased by 1.54 (1.55), 1.00 (1.00), 0.50 (0.50), 0.33 (0.33) times, respectively. The length of $|\mathcal{S}|$ increased because each node needs more time to charge its battery to its maximum level. Notice that $|\mathcal{S}|$ increases almost linearly when r_i is increased from one to 20. Further, the rate in which $|\mathcal{S}|$ increases in smaller networks, e.g., $|V| = 10$, is less than that of larger networks, e.g., $|V| = 50$. This is because more nodes mean more links need to be scheduled.

Effect on Charge/Discharge Cycles: For $|V| = 10$, 20, 30, 40, and 50, LSDBS produces a total charge/discharge cycles of 95, 396, 850, 1451, and 2325, respectively for all values of r_i. In particular, increasing r_i does not affect the number of battery charge/discharge cycles. This is because the number of charge/discharge cycles of battery at each node v_i that needs to accumulate $\mathcal{E}_i$ of energy, i.e., $\lceil \mathcal{E}_i / \tilde{b}_i^z \rceil$, depends only on the size of the battery's usable energy $\tilde{b}_i^z$. Thus, different values of r_i will only affect the time duration to harvest $\mathcal{E}_i$ of energy.

VI. Conclusion

This paper has proposed a novel link scheduling algorithm called LSDBS that considers nodes with a dual-batteries system and a battery cycle constraint. Our results show the benefits of using a dual-battery system in terms of schedule length and charge/discharge cycles. Specifically, as compared to using a single battery that has the same capacity and cycle constraint, LSDBS produces 35.19% (15%) less number of slots (charge/discharge cycles). Interestingly, LSDBS obtains 15.12% shorter schedules when compared to nodes that use a single battery, even without cycle constraint, and they have 87.13% fewer charge/discharge cycles.

References

[1] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards internet of things," in *IEEE 7th Annual Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2017, pp. 1–6.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] K. S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman, "Energy-harvesting wireless sensor networks (EH-WSNs): A review," *ACM Trans. Sensor Netw. (TOSN)*, vol. 14, no. 2, pp. 1–50, 2018.

[4] R. A. Huggins, "Mechanism of the memory effect in nickel electrodes," *Solid State Ionics*, vol. 177, no. 26, pp. 2643–2646, 2006.

[5] R. V. Bhat, M. Motani, C. R. Murthy, and R. Vaze, "Energy harvesting communications with batteries having cycle constraints," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 1, pp. 263–276, Mar. 2020.

[6] ——, "Energy harvesting communications using dual alternating batteries," *arXiv preprint arXiv:1801.03813*, 2018.

[7] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in wireless sensor networks: Distributed edge-coloring revisited," *J. Parallel Distrib. Comput.*, vol. 68, no. 8, pp. 1122–1134, 2008.

[8] G. Sun, G. Qiao, and L. Zhao, "Efficient link scheduling for rechargeable wireless ad hoc and sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 1, no. 1, pp. 1–14, Dec. 2013.

[9] T. Tony, S. Soh, K.-W. Chin, and M. Lazarescu, "Link scheduling in rechargeable wireless sensor networks with imperfect batteries," *IEEE Access*, vol. 7, pp. 104 721–104 736, 2019.

[10] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1384–1412, 2016.

[11] S. Guan, J. Zhang, Z. Song, B. Zhao, and Y. Li, "Energy-saving link scheduling in energy harvesting wireless multihop networks with the non-ideal battery," *IEEE Access*, vol. 8, pp. 144 027–144 038, 2020.

[12] T. Tony, S. Soh, K.-W. Chin, and M. Lazarescu, "Link scheduling in rechargeable wireless sensor networks with battery memory effects," in *ITNAC*, Melbourne, Australia, Nov. 2020, to appear.

[13] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, Mar. 1999.

[14] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact on interference on multi-hop wireless network performance," in *ACM MobiCom*, San Diego, CA, USA, Sep. 2003, pp. 66–80.

[15] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 910–917, 1988.