IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 11, 2021, accepted January 18, 2021, date of publication January 22, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053494

Link Scheduling in Rechargeable Wireless Sensor Networks With Imperfect Battery and Memory Effects

TONY TONY[1], (Member, IEEE), SIETENG SOH[1], (Member, IEEE), KWAN-WU CHIN[2], AND MIHAI LAZARESCU[1], (Member, IEEE)

[1]School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Perth, WA 6102, Australia
[2]School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2500, Australia

Corresponding author: Tony Tony (tony@postgrad.curtin.edu.au)

The work of Tony Tony was supported by the Indonesia Lecturer Scholarship (BUDI) from the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance, Indonesia.

ABSTRACT This paper considers the novel problem of deriving a Time Division Multiple Access (TDMA) link schedule for rechargeable wireless sensor networks (rWSNs). Unlike past works, it considers: (i) the energy harvesting time of nodes, (ii) a *battery cycle constraint* that is used to overcome so called *memory effects*, and (iii) battery imperfections, i.e., leakage. This paper shows analytically that the battery cycle constraint and leaking batteries lead to unscheduled links. Further, it presents a greedy heuristic that schedules links according to when their corresponding nodes have sufficient energy. Our simulations show that enforcing the battery cycle constraint increases the link schedule by up to 1.71 (0.31) times for nodes equipped with a leaking (leak-free) battery. When nodes have a leaking battery, the derived schedules are on average 1.05 times longer than the case where nodes have a leak-free battery. Finally, the battery cycle constraint reduces the number of charge/discharge cycles by up to 47.41% (45.67)% when nodes have a leak (leak-free) battery. Between leak-free and leak battery scenarios, using the former produces up to 51.46% fewer cycles than the latter.

INDEX TERMS Battery cycle constraint, harvesting time, link schedule, wireless sensor networks.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) form the sensing layer of Internet of Things (IoTs) [1]. They have been used for environmental or habitat monitoring [2]–[4]. They are also widely used in various industries such as manufacturing [5]. A well-known issue faced by WSNs is that nodes have limited energy. In many applications, it is impractical to replace the batteries of nodes, especially when there is a large number of sensor nodes and they are deployed in difficult-to-reach locations.

To this end, rechargeable WSNs (rWSNs) are now of great interest because sensor nodes are able to harvest energy from their environment, e.g., sunlight. However, nodes may experience time-varying energy arrivals, meaning when a node exhausts its energy, it will have to spend time harvesting energy before it is able to continue executing tasks. The time used to harvest a given amount of energy is affected by the type of energy source as well as a node's location [6]. For instance, assume a solar panel has a power density of $15,000\ \mu\text{W/cm}^3$ and $20\ \mu\text{W/cm}^3$ for outdoor and indoor settings, respectively [7]. Hence, a node with a $50\ \text{cm}^3$ solar panel will have a corresponding energy harvesting rate of 300 mJ/s (outdoor) and 0.4 mJ/s (indoor). Assuming a Mica2 mote [8], which requires 30 mJ of energy to transmit/receive a packet, it will need to spend 0.1s (outdoor) or 75s (indoor) harvesting energy before it can transmit/receive one packet.

Another important issue of interest recently is the lifetime of rechargeable batteries. Among others, one factor that affects the lifetime of batteries is *memory effects* [9], which decrease their usable capacity if they are charged and discharged repeatedly after a partial discharge and charge, respectively. Another factor is the percentage of discharged energy relative to a battery's overall capacity, which is also called the battery's *Depth of Discharge (DoD)* [10]. Further, frequent battery charge and discharge affect a battery's lifetime [11]. To prolong the lifetime of a battery, we can impose

The associate editor coordinating the review of this manuscript and approving it for publication was Mahdi Zareei.

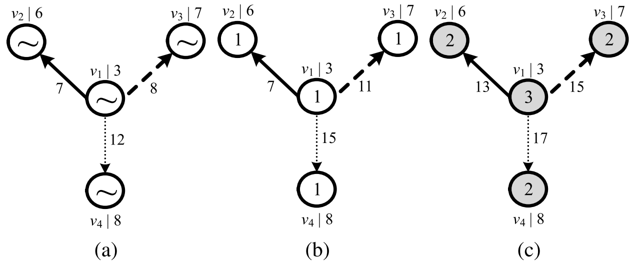


FIGURE 1. **An example (a) with interference and harvesting time, (b) plus battery capacity, and (c) plus battery cycle constraints. The number inside each node shows the capacity of its battery. The number next to each link denotes its activation time, and $v_x|z$ represents node x requiring z time slots to accumulate one unit of energy.**

a *battery cycle* constraint, i.e., a node must charge (discharge) its battery completely before fully discharging (charging) its battery again [12].

A fundamental problem in rWSNs is channel access or *link scheduling* [13]. This paper considers a rWSN that uses a Time Division Multiple Access (TDMA) link schedule; this ensures nodes do not experience collisions, which waste energy, and nodes only need to be active during their allocated time slot. A link scheduler allocates links into slot(s), and links in each slot do not suffer from excessive interference. Ideally each slot should have a high number of links, which ensures a high network capacity. Moreover, a short schedule means a link transmits frequently.

This paper considers a *novel* research aim: derive a short TDMA link schedule that considers nodes with varying energy harvesting rates and battery memory effects. To achieve this aim, solutions must address a number of problems. Consider Figure 1a, 1b, and 1c. Note that links (v_1, v_2), (v_1, v_3), and (v_1, v_4) interfere with each other and thus cannot be scheduled concurrently. Consider the Harvest-Store-Use (HSU) battery charging model [6], where the harvested energy in slot t can only be used in slot $t+1, t+2, \cdots$. Node v_1 needs to wait for three time slots to accumulate one unit of energy, denoted as $v_1|3$, and this energy can only be used after slot $t = 3$. First, consider the case where nodes have unlimited battery capacity; denoted by $\sim$ in Figure 1a. Node v_1 can use its stored energy in time slot $t = 4$. However, none of its links can be scheduled at time 4 because its neighbors have insufficient energy to receive a packet. For example, link (v_1, v_2) can be scheduled no earlier than at slot $6 + 1 = 7$. After node v_1 transmits a packet to node v_2 at time $t = 7$, its remaining energy is sufficient to transmit a packet to node v_3. Thus, link (v_1, v_3) is scheduled at time $t = 8$. Then, node v_1 needs to accumulate energy before it can transmit a packet to node v_4. Thus, link (v_1, v_4) is scheduled at slot $t = 8 + 3 + 1 = 12$, producing the schedule length of 12.

Next, consider the case where each battery has a capacity of one unit; see Figure 1b. This means the battery at node v_1 can be recharged only after it is used at time $t = 7$. Thus, node v_1 can transmit the second packet no earlier than at time $t = 7+3+1 = 11$, i.e., after it has harvested sufficient energy. Further, it can transmit the third packet no earlier than time $11 + 4 = 15$. Hence, the schedule length is 15.

Lastly, consider the case where each battery has a *battery cycle* constraint; see Figure 1c. Node v_1 has a battery with three units of energy and node v_2, v_3, and v_4 have a battery with two units of capacity. Thus, v_1 needs to wait until slot $t = 3 \times 3 + 1 = 10$ to fully recharge its battery before it can transmit one packet. However, it cannot do so because its neighbors' battery is yet to be fully recharged. That is node v_2, v_3, and v_4 have to wait until slot $t = 6 \times 2 + 1 = 13$, $t = 15$, and $t = 17$, respectively before their battery can be discharged. The schedule length in this case is 17.

Given the above research aim, this paper makes the following *contributions*:

- It proposes a TDMA link scheduler that considers (i) sensor nodes with a different energy harvesting rate and finite battery capacity, (ii) batteries operation governed by a *battery cycle* constraint, (iii) batteries with a leakage rate and storage efficiency, and (iv) activating each link (i, j) at least $w_{i,j}$ times.
- It develops a *novel* heuristic technique that greedily schedules links that can be activated according to when their end-nodes are able to transmit/receive a packet.
- It analytically shows that some links cannot be scheduled for networks that contain batteries with non-negative leakage rate. This conclusion is supported by our simulation results in Section IV-B.

Except for references [14]–[17], and [18], there is no research that focuses on link scheduling where nodes require varying amount of time to harvest energy. The authors in [17] proposed three link scheduling algorithms to activate links with the maximum weight. The weight of a link represents the amount of consumed energy when it is active. They aim to minimize the amount of stored energy and reduce energy waste. Sun *et al.* [14] propose two link schedulers that correspond to links with and without link weight. They aim to maximize network throughput and use the HSU [6] battery charging model, where a node must first store before using its harvested energy. Recently, the authors of [15] consider the Harvest-Use-Store (HUS) [19] battery usage protocol, which uses a super capacitor to store harvested energy. This allows a node to use its harvested energy immediately as well as allowing it to store any remaining energy for later use. In a subsequent work, in [16], the authors consider imperfect batteries that leak and have storage inefficiency. Each battery has a recharging time that defines when a node has sufficient energy for one packet data transmission/reception. The authors of [14] consider infinite battery capacity whereas reference [15] and [16] have a limited battery size. These works reported in [14], [15], and [16] consider rechargeable battery with *shallow* recharge or *partial* recharge and/or discharge [6]. As reported in [11], shallow charging of a battery causes *memory effects* that reduce a battery's lifetime, which ultimately reduces the lifetime of a rWSN; such effects are not considered in [14]–[16], and [17]. Specifically, the authors of [14] and [15] consider only battery recharging time and battery capacity. Tony *et al.* [16] extend the work in [15] by including battery leakage and storage efficiency in their system model. The authors in [17] considered only the battery storage efficiency.

To this end, there are no prior works on link scheduling for rWSNs that aim to avoid battery memory effects via a battery cycle constraint. Recently, only reference [18] that considers the battery cycle constraint. However, it considers only leak-free battery. Henceforth, this paper extends the work in [18] to include all of the following factors: battery recharging time, capacity, leakage, storage efficiency, and cycle constraint.

We summarize the differences between this paper and the previous work [16] as follows:

1) This paper imposes a battery cycle constraint to alleviate memory effects [20]. The work in [16] does not consider the said constraint. As discussed in [20] and Section I, memory effects shorten battery lifetime and reduce battery capacity, and thus the work in this paper is important to increase the operational lifetime of rWSNs.
2) This paper uses the HSU [19] battery charging model, while the previous work [16] considers the HUS model [19]. Note that it is possible to revise the HSU model to apply in the HUS model.
3) Due to differences 1) and 2), the proposed algorithm, i.e., LSBCC in this paper is different from algorithm LS-rWSN in [16]. Further, this paper analytically shows that the battery cycle constraint and battery with leakage rate $\mu_i > 0$ may lead to unscheduled links. The paper also shows the trade-off between a longer superframe length. It reduces the number of charging/discharging cycles that affect battery quality in terms of lifetime and capacity when imposing battery cycle constraint.

The rest of this paper is organized as follows. Section II contains the network model and problem at hand. Our solution is described in Section III, and its performance evaluation is reported in Section IV. Finally, Section V concludes the paper and provides future research directions.

II. PRELIMINARIES

We first formalize our rWSN model; all key notations are summarized in Table 1. After that, Section II-B formalizes the problem of interest.

A. NETWORK MODEL

A rWSN is modeled 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/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 at least $w_{i,j}$ times in the generated schedule; e.g., in Figure 3a, we see $w_{2,4} = 3$. Let ϵ (in Joule) be the energy consumed to transmit or receive one packet. We assume equal energy usage for transmission and reception.

TABLE 1. Notations and definitions.

Notation	*Definition*
V	The set of sensor nodes.
v_i	A sensor node i.
E	The set of directional links.
$l_{i,j}$	A directed link from v_i to v_j.
$w_{i,j}$	The link weight for each incident link $l_{i,j}$ at node v_i.
$\mathcal{R}_i$	The transmission range of each node v_i.
r_i	Time for the harvester in node v_i to accumulate 1ϵ units of energy.
b_i	The battery capacity of node v_i.
$\tilde{b}_{i,t}$	The energy level of the battery in charging mode at node v_i at time slot t.
$b_{i,t}$	The energy level of the battery in discharging mode at node v_i at time slot t.
$b_{i,max}$	The upper limit capacity of battery at node v_i.
$b_{i,min}$	The lower limit capacity of battery at node v_i.
μ_i	The battery leakage rate of node v_i.
η_i	The battery storage efficiency of node v_i.
$\tilde{t}^+_{i,k}$	The start time of charging cycle k of the battery at node v_i.
$\tilde{t}^-_{i,k}$	The end time of charging cycle k of the battery at node v_i.
$t^+_{i,k}$	The start time of discharging cycle k of the battery at node v_i.
$t^-_{i,k}$	The end time of discharging cycle k of the battery at node v_i.
$\tilde{\tau}_{i,k}$	The charging time interval of the battery at node v_i at cycle $k \geq 1$.
$\tau_{i,k}$	The discharging time interval of the battery at node v_i at cycle $k \geq 1$.
t_i	The most recent time when the battery at node v_i was discharged.
b_{i,t_i}	The energy level at node v_i at time slot t_i.
$t_{i,j}$	The earliest time in which the battery at end nodes of link (i, j) can be discharged.
σ_{i,t_i}	A binary variable that indicates whether the battery at node v_i can be discharged at time t_i.
T_i	The earliest time slot when the battery at node v_i can be discharged to transmit/receive a packet.
$\mathcal{T}_{i,k}$	The latest time slot when the battery at node v_i can be discharged in a discharging cycle k.
$\mathcal{S}$	The superframe or schedule length.
ϵ	Energy needed to transmit/receive a packet (in Joules).

This paper uses the protocol interference model [21], which considers (i) primary interference, where each node is half-duplex, and (ii) secondary interference, where a receiver, say A, that is receiving a packet from a transmitter, say B, is interfered by another transmitter, say C. The interference between links is modeled by a conflict graph $C_G(V', E')$ [22], 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.

A TDMA superframe or a link schedule consists of equal sized time slots. 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 an energy outage.

A node v_i is equipped with a harvester that scavenges energy from its environment, e.g., solar, and it is equipped with a rechargeable battery with capacity b_i (in unit of ϵ).

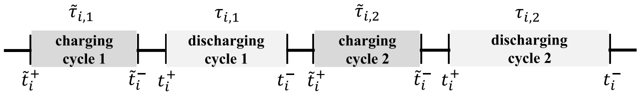


FIGURE 2. Charge-discharge cycles at node v_i at cycle $k = 1$ and $k = 2$.

It uses the HSU battery charging model [6], where the harvested energy at time slot t must first be stored before it is used in later slots. Note that this paper makes no assumption about the energy harvesting model used by nodes. That is, it assumes each node has a known energy arrival rate. Let $r_i \geq 1$ (in slots) be the *harvesting time* or 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. Let $0 < \eta_i \leq 1$ be the storage efficiency and $0 \leq \mu_i < 1$ be the battery leakage factor (per time slot) of node v_i. This work omits the following cases. First, when $\eta_i = 0$ and $\mu_i = 1$, the battery of nodes cannot store any harvested energy and retain its energy, respectively. Second, in each slot, the amount of harvested energy must be larger than the battery leakage rate μ_i. Otherwise, any harvested energy will be lost immediately due to battery leakage. In both cases, nodes will have no energy to activate links.

All nodes have a minimum battery capacity, i.e., for node v_i, we have $b_i \geq 1\epsilon$. They have a single rechargeable battery with a so called *battery cycle* constraint [20]. This constraint requires the battery of node v_i to be (i) charged to its maximum capacity $b_{i,max}$ before it can be used/discharged, for $2 \leq b_{i,max} \leq b_i$, and (ii) discharged to its minimum capacity, $b_{i,min} \geq 1$, before it can be charged. We call (i) and (ii) respectively as the discharging and charging constraint. Consequently, the battery at each node v_i can be in one of two modes: (i) *charging*, or (ii) *discharging*. More specifically, the battery cannot be in the charging and discharging mode at the same time. Without loss of generality, we assume each battery has an initial energy level of $b_{i,min}$. Further, we assume $b_{i,min}$ and $b_{i,max}$ are integers, where $b_{i,min} < b_{i,max}$.

For node v_i, let $\tilde{t}^+_{i,k}$ and $\tilde{t}^-_{i,k}$ be respectively the start and end time of its k-th charging cycle. Similarly, for discharging cycle k at node v_i, its start and end time are denoted respectively as $t^+_{i,k}$ and $t^-_{i,k}$. Further, $\tilde{\tau}_{i,k}$ and $\tau_{i,k}$ respectively are the *charging* and the *discharging* time interval of the battery at node v_i in cycle $k \geq 1$; these quantities are computed as $\tilde{\tau}_{i,k} = \tilde{t}^-_{i,k} - \tilde{t}^+_{i,k}$ and $\tau_{i,k} = t^-_{i,k} - t^+_{i,k}$. In other words, the battery at node v_i is being charged during time interval $\tilde{\tau}_{i,k}$ and being discharged during time interval $\tau_{i,k}$ for each cycle k. As illustrated in Figure 2, each battery follows a sequence of charge-discharge cycle. Thus, we have $t^+_{i,k} = \tilde{t}^-_{i,k} + 1$ and $\tilde{t}^+_{i,k} = t^-_{i,k} + 1$. Note that, for each cycle k, the value of $\tilde{\tau}_{i,k}$ is dependent on r_i, η_i as well as $b_{i,max}$, while the length of $\tau_{i,k}$ is affected by $b_{i,min}$ and the number of times the battery is used to transmit/receive packets at each cycle k, denoted by $u_{i,k}$. In addition, both times are affected by the battery's leakage factor μ_i. Note that as r_i, μ_i, η_i, and $b_{i,max}$ are constants, all intervals $\tilde{\tau}_{i,k}$ have equal length. In contrast, the value of $\tau_{i,k}$ may vary at different cycles because $u_{i,k}$ varies according to the number of transmitted/received packets. In the remainder of the paper, the cycle number k is omitted if the context is clear.

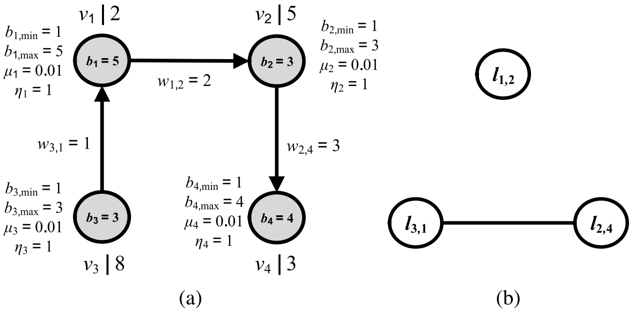


FIGURE 3. A rWSN as a (a) graph G, and its (b) conflict graph C_G.

Let $\tilde{b}_{i,t}$ and $b_{i,t}$ (in unit of ϵ) denote the energy level of the battery at node v_i during a charging and discharging cycle at time slot t, respectively. Thus, we have $\tilde{b}_{i,\tilde{t}^-_i} = b_{i,max}$ and $b_{i,t^-_i} = b_{i,min}$. The battery level of node v_i at the beginning of each *charging* cycle, i.e., at time $\tilde{t}^+_i$, is

$$\tilde{b}_{i,\tilde{t}^+_i} = b_{i,min} - \mu_i b_{i,min} = (1 - \mu_i) b_{i,min}. \quad (1)$$

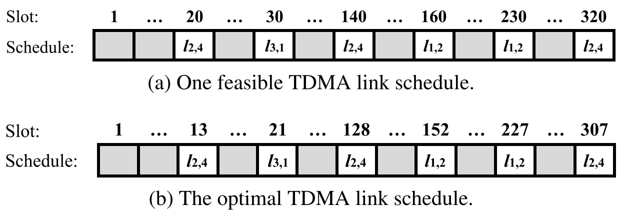
The reason is because (i) the battery stops discharging at time $t^-_i = \tilde{t}^+_i - 1$ when the amount of energy reaches $b_{i,min}$, and (ii) there is energy leakage of $\mu_i b_{i,min}$ from time $\tilde{t}^+_i - 1$ to $\tilde{t}^+_i$. On the other hand, the battery level of node v_i at the start of *discharging* cycle t^+_i is

$$b_{i,t^+_i} = b_{i,max} - \mu_i b_{i,max} = (1 - \mu_i) b_{i,max}. \quad (2)$$

The reason is because (i) the battery stops charging at time $\tilde{t}^-_i = t^+_i - 1$ when its battery level reaches its maximum capacity, i.e., $b_{i,max}$, and (ii) there is energy leakage of $\mu_i b_{i,max}$ from time $t^+_i - 1$ to t^+_i.

Let T_i be the earliest time slot when the battery at node v_i is in discharging mode. The earliest time in which link $l_{i,j}$ can be scheduled is at time $t_{i,j} = \mathbf{max}(T_i, T_j)$, i.e., when the end nodes of link $l_{i,j}$ can discharge their battery. For each node v_i, we initialize T_i to $\tilde{\tau}_i + 1$. It is updated when the battery at node v_i is discharged to transmit/receive a packet.

Figure 3a presents a rWSN with four nodes to illustrate our network model. It shows values of $b_{i,min}$, $b_{i,max}$, b_i, μ_i, η_i, and r_i for each node v_i as well as the weight of each link $l_{i,j}$. The battery level of each node v_i in charging and discharging mode is respectively computed using Eq. (1) and (2), e.g., $b_{1,t^+_1} = b_{1,max} \times \hat{\mu_1} = 5 \times 0.99 = 4.95$ and $\tilde{b}_{1,\tilde{t}^+_1} = b_{1,min} \times \hat{\mu_1} = 1 \times 0.99 = 0.99$. As shown later in Section III, one can compute the charging time and discharging time intervals for each battery to obtain $\tilde{\tau}_1 = 9$, $\tilde{\tau}_2 = 12$, $\tilde{\tau}_3 = 20$, and $\tilde{\tau}_4 = 10$. Thus, we have $T_1 = \tilde{\tau}_1 + 1 = 10$, $T_2 = 13$, $T_3 = 21$, and $T_4 = 11$. Therefore, the earliest time each link $l_{i,j}$ can be scheduled is computed as $t_{1,2} = \mathbf{max}(10, 13) = 13$, $t_{2,4} = 13$, $t_{3,1} = 21$. In Figure 3b, there are two links that experience primary interference, namely, link $l_{3,1}$ with $l_{1,2}$, and $l_{1,2}$ with $l_{2,4}$. Also shown is secondary interference at node v_1 that is caused by v_2, i.e., link $l_{3,1}$ with $l_{2,4}$.



(a) One feasible TDMA link schedule.

(b) The optimal TDMA link schedule.

FIGURE 4. TDMA link schedules for the rWSN in Figure 3 with a battery cycle constraint. Gray colored slots show no transmissions/receptions. Note that the figure shows only non-empty slots with empty slot represented as "...".

B. PROBLEM STATEMENT

Given a rWSN, our problem, called **Link Scheduling Memory Effects (LSME)**, is to generate the TDMA link schedule $\mathcal{S}$ with minimum length $|\mathcal{S}|$ that satisfies the following constraints: (i) the battery at each node $v_i \in V$ satisfies the *battery cycle constraint*, (ii) each link $l_{i,j} \in E$ can be scheduled at time slot t only if its end nodes are in discharging mode, (iii) each link $l_{i,j} \in E$ is scheduled at least $w_{i,j}$ times in $\mathcal{S}$.

Figure 4 shows two example link schedules. Figure 4a presents a *feasible* link schedule with 320 slots, which satisfies constraints (i), (ii), and (iii). Figure 4b gives a shorter *feasible* schedule with $|\mathcal{S}| = 307$ slots. Note that link scheduling is known to be NP-hard, where works such as [22] and [23], assume nodes with unlimited energy. This is simply a special case of **LSME**. Thus, our problem is also NP-hard.

III. SOLUTION

This section first describes eight propositions relied upon by our greedy algorithm. It then provides two propositions to show that **LSME** might not have a feasible solution when the battery of nodes has a non-negative leakage rate; i.e., $\mu_i > 0$. Finally, the details of the proposed greedy algorithm are presented in Section III-D.

A. KEY PROPERTIES

Proposition 1 and 2 relate to batteries in a charging cycle, and Proposition 3 to 8 relate to batteries in a discharging cycle. For brevity, we define $\hat{r}_i = r_i/\eta_i$, and $\hat{\mu}_i = 1 - \mu_i$.

Proposition 1: The energy level (in ϵ) of a battery in the charging mode at node v_i at time slot t, for $\tilde{t}_i^+ \le t \le \tilde{t}_i^-$, is

$$\tilde{b}_{i,t} = \mathbf{min}\left(b_{i,max}, b_{i,min} \times \hat{\mu}_i^{\,t-\tilde{t}_i^+ +1} + \sum_{p=0}^{t-\tilde{t}_i^+ -1} \frac{\hat{\mu}_i^{\,p}}{\hat{r}_i}\right). \quad (3)$$

Proof: The stored energy at time $\tilde{t}_i^+ + 1$ is computed by subtracting the energy lost due to battery leakage, i.e., $\mu_i \tilde{b}_{i,\tilde{t}_i^+}$ plus the energy harvested at slot $\tilde{t}_i^+$, i.e., $1/\hat{r}_i$. Thus, $\tilde{b}_{i,\tilde{t}_i^+ +1} = (1-\mu_i)\tilde{b}_{i,\tilde{t}_i^+} + 1/\hat{r}_i$. The energy level $\tilde{b}_{i,\tilde{t}_i^+ +2}$ is computed from $\tilde{b}_{i,\tilde{t}_i^+ +1}$ by (i) subtracting the energy leakage that occurs from time $\tilde{t}_i^+ + 1$ to $\tilde{t}_i^+ + 2$, (ii) adding the energy harvested in time slot $\tilde{t}_i^+ + 1$, and (iii) substituting $\tilde{b}_{i,\tilde{t}_i^+ +1}$ with $(1-\mu_i)\tilde{b}_{i,\tilde{t}_i^+} + 1/\hat{r}_i$. Steps (i) and (ii) obtain $\tilde{b}_{i,\tilde{t}_i^+ +2} = (1-\mu_i)\tilde{b}_{i,\tilde{t}_i^+ +1} + 1/\hat{r}_i$. Step (iii) substitutes $\tilde{b}_{i,\tilde{t}_i^+ +1}$ with $(1-\mu_i)\tilde{b}_{i,\tilde{t}_i^+} + 1/\hat{r}_i$ to produce $\tilde{b}_{i,\tilde{t}_i^+ +2} = (1-\mu_i)[(1-\mu_i)\tilde{b}_{i,\tilde{t}_i^+}] + 1/\hat{r}_i = (1-\mu_i)^2\tilde{b}_{i,\tilde{t}_i^+} + (1-\mu_i)/\hat{r}_i + 1/\hat{r}_i$. Then, using energy level $\tilde{b}_{i,\tilde{t}_i^+ +2}$, steps (i) to (iii) are used to compute energy level at time $\tilde{t}_i^+ + 3$, i.e., $\tilde{b}_{i,\tilde{t}_i^+ +3} = (1-\mu_i)^3\tilde{b}_{i,\tilde{t}_i^+} + (1-\mu_i)^2/\hat{r}_i + (1-\mu_i)/\hat{r}_i + 1/\hat{r}_i$. Repeating steps (i) to (iii) for time $\tilde{t}_i^+ + 4, \ldots, t-1, t$, we obtain the stored energy at the beginning of time t, i.e., $\tilde{b}_{i,t} = (1-\mu_i)^{t-\tilde{t}_i^+}\tilde{b}_{i,\tilde{t}_i^+} + \sum_{p=0}^{t-\tilde{t}_i^+ -1}(1-\mu_i)^p/\hat{r}_i$. Substituting $(1-\mu_i)b_{i,min}$ in Eq. (1) for $\tilde{b}_{i,\tilde{t}_i^+}$, we have $\tilde{b}_{i,t} = (1-\mu_i)^{t-\tilde{t}_i^+ +1}b_{i,min} + \sum_{p=0}^{t-\tilde{t}_i^+ -1}\frac{(1-\mu_i)^p}{\hat{r}_i}$. However, $\tilde{b}_{i,t}$ is bounded by the upper limit of battery capacity $b_{i,max}$, which implies $\tilde{b}_{i,t} \le b_{i,max}$. Substituting $(1-\mu_i)$ with $\hat{\mu}_i$, we obtain the expression stated in the proposition. □

Note that when the battery has 100% storage efficiency and no leakage, i.e., $\eta_i = 1$ and $\mu_i = 0$, respectively, Eq. (3) reduces to

$$\tilde{b}_{i,t} = \mathbf{min}(b_{i,max}, b_{i,min} + (t - \tilde{t}_i^+)/r_i). \quad (4)$$

Proposition 2 computes the charging time interval $\tilde{\tau}_i = \tilde{t}_i^- - \tilde{t}_i^+$ of the battery at node v_i, i.e., the number of slots needed to charge the battery to its maximum level $b_{i,max}$ starting from its minimum level $b_{i,min}$.

Proposition 2: The charging time interval for the battery of node v_i is computed as

$$\tilde{\tau}_i = \begin{cases} r_i(b_{i,max} - b_{i,min}), & \eta_i = 1 \; and \; \mu_i = 0 \\ \left\lceil \dfrac{log\,(\frac{1-\hat{r}_i\mu_i b_{i,max}}{1-\hat{r}_i\mu_i\hat{\mu}_i b_{i,min}})}{log\,\hat{\mu}_i} \right\rceil, & 0 < \eta_i < 1 \; and \; 0 < \mu_i < 1. \end{cases} \quad (5)$$

Proof: We set $t = \tilde{t}_i^-$ in Eq. (3) to compute the maximum energy level of the battery at node v_i, i.e., $b_{i,max}$. Thus, we have

$$b_{i,max} = b_{i,min} \times \hat{\mu}_i^{\,\tilde{\tau}_i+1} + \sum_{p=0}^{\tilde{\tau}_i-1}\frac{\hat{\mu}_i^{\,p}}{\hat{r}_i}. \quad (6)$$

For case (i): $\eta_i = 1$ and $\mu_i = 0$, we set $\hat{\mu}_i = 1$ and $\hat{r}_i = r_i$ in Eq. (6) to yield $b_{i,max} = b_{i,min} + \tilde{\tau}_i/r_i$. Thus, $\tilde{\tau}_i = r_i(b_{i,max} - b_{i,min})$ as shown in Eq. (5). For case (ii): $0 < \eta_i < 1$ and $0 < \mu_i < 1$, i.e., $\hat{\mu}_i \neq 1$, we use the geometric series [24] to produce $\sum_{p=0}^{\tilde{\tau}_i-1}\frac{\hat{\mu}_i^{\,p}}{\hat{r}_i} = \frac{1}{\hat{r}_i}\frac{\hat{\mu}_i^{\tilde{\tau}_i}-1}{\hat{\mu}_i-1}$. Thus, Eq. (6) becomes

$$b_{i,max} = b_{i,min} \times \hat{\mu}_i^{\,\tilde{\tau}_i+1} + \frac{1}{\hat{r}_i}\frac{\hat{\mu}_i^{\tilde{\tau}_i} - 1}{\hat{\mu}_i - 1}. \quad (7)$$

Solving Eq. (7) for $\tilde{\tau}_i$, we obtain Eq. (5). □

The amount of energy that leaks from the battery of node v_i per slot, i.e., $\hat{r}_i\mu_i$, and battery's leakage rate μ_i in case (ii) of Eq. (5), must be less than $\frac{1}{b_{i,max}}$ and $\frac{1}{\hat{r}_i\hat{\mu}_i b_{i,min}}$, respectively. Otherwise, the battery will never be able to charge to its maximum level.

We first present Eq. (8) and (9) that are used in Proposition 3 and 4. Consider a battery of node v_i with energy level of x at time t_1, i.e., $b_{i,t_1} = x$. As the battery leaks, energy level of the battery at time $t_2 \ge t_1$ is reduced to

$$b_{i,t_2} = x\hat{\mu}_i^{\,t_2-t_1}. \quad (8)$$

Eq. (8) is used to compute the amount of energy that has leaked from the battery of node v_i from time t_1 to t_2, i.e., $\Delta_{i,t_1}^{t_2} = b_{i,t_1} - b_{i,t_2}$ for $b_{i,t_1} = x$ as

$$\Delta_{i,t_1}^{t_2} = x(1 - \hat{\mu}_i^{t_2 - t_1}). \tag{9}$$

Proposition 3 to 8 relate to a battery in discharging mode. Proposition 3 computes energy level of the discharging battery at time t, for $t_i < t \leq t_i^-$, where t_i is the most recent time the battery was used. From time t_i to t, the energy level decreases due to battery leakage *only*. Thus, Proposition 3 is valid only for batteries with a non-negative leakage rate; i.e., $\mu_i > 0$.

Proposition 3: The energy level (in ϵ) of the discharging battery at node v_i at time slot t, for $t_i < t \leq t_i^-$ is

$$b_{i,t} = \mathbf{max}(b_{i,min}, b_{i,max} \times \hat{\mu}_i^{t-t_i^+ +1} - \hat{\mu}_i^{t-t_i}). \tag{10}$$

Proof: Following Eq. (2), the battery level at time t_i^+ is $b_{i,t_i^+} = b_{i,max} \times \hat{\mu}_i$. Using Eq. (9), the amount of energy leaked from time t_i^+ to t_i, for $x = b_{i,t_i^+}$ is

$$\Delta_{i,t_i^+}^{t_i} = b_{i,t_i^+} \times (1 - \hat{\mu}_i^{t_i - t_i^+}). \tag{11}$$

Thus, after spending 1ϵ of energy at time t_i, the remaining energy of the battery at the end of slot t_i is

$$b_{i,t_i} = b_{i,t_i^+} - \Delta_{i,t_i^+}^{t_i} - 1. \tag{12}$$

Now, as per Eq. (9), the energy leaked from time t_i to t, for $x = b_{i,t_i}$, is

$$\Delta_{i,t_i}^{t} = b_{i,t_i} \times (1 - \hat{\mu}_i^{t-t_i}). \tag{13}$$

Thus, the remaining energy at time t is

$$b_{i,t} = b_{i,t_i} - \Delta_{i,t_i}^{t}. \tag{14}$$

Using (2) in (11), (2) and (11) in (13), (12) in (13), and (12) and (13) in (14), we obtain

$$b_{i,t} = b_{i,max} \times \hat{\mu}_i^{t-t_i^+ +1} - \hat{\mu}_i^{t-t_i}. \tag{15}$$

As the energy level is lower bounded by $b_{i,min}$, we thus obtain Eq. (10). □

Note that when the leakage rate is $\mu_i = 0$, the energy level of a battery at node v_i at time slot t is $b_{i,t} = b_{i,t_i} - 1$. This is because the energy level decreases only due to packet transmission/reception.

Let $\delta_{i,b_1}^{b_2}$ be number of time slots needed to discharge the battery of node v_i from level $b_1 \leq b_{i,max}$ to $b_2 \geq b_{i,min}$ due to battery leakage *only*. The following Proposition 4 computes $\delta_{i,b_1}^{b_2}$, which is applicable only when each battery has non-negative leakage rate $\mu_i > 0$.

Proposition 4: The discharging time interval, due to battery leakage only, for the battery of node v_i, to decrease from $b_1 \leq b_{i,max}$ to $b_2 \geq b_{i,min}$ is given as

$$\delta_{i,b_1}^{b_2} = \lfloor \frac{log(b_2) - log(b_1)}{log(\hat{\mu}_i)} \rfloor. \tag{16}$$

Proof: First, we set $\Delta_{i,t_1}^{t_2} = b_1 - b_2$ and $x = b_1$ in Eq. (9). As a result, we have

$$b_1 - b_2 = b_1(1 - \hat{\mu}_i^{\delta_{i,b_1}^{b_2}}). \tag{17}$$

Then we solve Eq. (17) for $\delta_{i,b_1}^{b_2}$ to obtain Eq. (16). □

Proposition 5 computes the next earliest time when the battery at node v_i can be used to transmit/receive a packet.

Proposition 5: Consider the battery at node v_i is lastly used at time $t_i \geq \tilde{\tau}_i + 1$. The next earliest time slot when the battery at node v_i can be used to transmit/receive a packet is

$$T_i = \begin{cases} t_i + \sigma_{i,t_i} \times \tilde{\tau}_i + 1, & \mu_i = 0 \\ t_i + \sigma_{i,t_i} \times (\delta_{i,b_1}^{b_2} + \tilde{\tau}_i) + 1, & \mu_i > 0. \end{cases} \tag{18}$$

Proof: The next earliest time the battery at node v_i can be used depends on the remaining battery level at time t_i, i.e., b_{i,t_i}. For Case (i): $\mu_i = 0$, Eq. (18) considers two sub-cases: (a) $b_{i,t_i} = b_{i,min}$, and (b) $b_{i,t_i} > b_{i,min}$. For sub-case (a), the battery needs to be recharged. This sub-case requires charging time interval of $\tilde{\tau}_i$, computed by Eq. (5). Thus, we have $T_i = t_i + \tilde{\tau}_i + 1$ because the harvested energy needs to be stored first before it can be used. For this sub-case, σ_{i,t_i} in Eq. (18) is set to 1. For sub-case (b), the battery at node v_i still can be discharged to transmit/receive another packet at time $t + 1$, i.e., $T_i = t_i + 1$, and thus σ_{i,t_i} in Eq. (18) is set to 0.

For Case (ii): $\mu_i > 0$, there are two sub-cases: (a) $b_{i,t_i} < (b_{i,min} + 1)$, and (b) $b_{i,t_i} \geq (b_{i,min} + 1)$. For sub-case (a), when the remaining energy is less than $(b_{i,min} + 1)$, the battery needs to be recharged. However, the battery cycle constraint requires the energy level of the battery to reach $b_{i,min}$ first before it can be recharged. Using Eq. (16), it takes $\lfloor \frac{log(b_{i,min}) - log(\ b_{i,t_i})}{log(\hat{\mu}_i)} \rfloor$ slots to discharge the battery (due to leaking) from level $b_{i,t_i} = b_1$ to $b_{i,min} = b_2$. Then, we use Eq. (5) to compute the charging time interval, $\tilde{\tau}_i$. Thus, we have $T_i = t_i + (\delta_{i,b_1}^{b_2} + \tilde{\tau}_i) + 1$ because the harvested energy needs to be stored first before it can be used. For this sub-case, we have $\sigma_{i,t_i} = 1$. For sub-case (b), the battery at node v_i can be discharged to transmit/receive another packet at time t_i. Thus, the next earliest time the battery at node v_i can be discharged to transmit/receive a packet is slot $T_i = t_i + 1$. For this sub-case, we have $\sigma_{i,t_i} = 0$. □

For a given discharging cycle k, let $\mathcal{T}_{i,k} > T_i$ be the time at which the energy level at node v_i is $b_{min} + 1$. In other words, $\mathcal{T}_{i,k}$ is the latest time the battery can be used in the discharging cycle k.

Proposition 6: The latest time the battery can be used for a given discharging cycle k is

$$\mathcal{T}_{i,k} = T_i + \lfloor \frac{log(b_{i,min} + 1) - log(b_{i,T_i})}{log(\hat{\mu}_i)} \rfloor. \tag{19}$$

Proof: The energy level at time T_i is b_{i,T_i}. The discharging time interval from level b_{i,T_i} to $b_{i,min} + 1$ can be computed using Eq. (16) for $b_1 = b_{i,T_i}$ and $b_2 = b_{i,min} + 1$. Thus, time $\mathcal{T}_{i,k}$ can then be obtained by computing the sum of time T_i and the discharging time, as given in Eq. (19). □

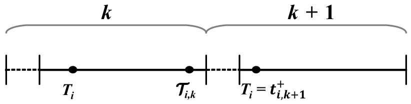


FIGURE 5. An Illustration for Proposition 6 and 7. The dashed (solid) lines represent charging (discharging) cycles.

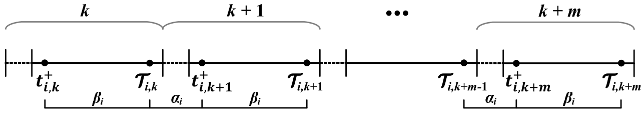


FIGURE 6. An illustration for Eq. (21) to Eq. (24).

For a given time $\mathcal{T}_{i,k}$, Proposition 7 computes the earliest time T_i at which the battery at node v_i can be used to transmit/receive a packet. In other words, $T_i = t^+_{i,k+1}$. Figure 5 illustrates the relationship between times $\mathcal{T}_{i,k}$ and T_i of Proposition 6 and 7, respectively.

Proposition 7: The earliest time $T_i > \mathcal{T}_{i,k}$ at which the battery at node v_i can be used is

$$T_i = \mathcal{T}_{i,k} + \lfloor \frac{log\ b_{i,min} - log\ b_{i,\mathcal{T}_{i,k}}}{log\ \hat{\mu}_i} \rfloor + \tilde{\tau}_i + 1. \tag{20}$$

Proof: At time $\mathcal{T}_{i,k}$, node v_i's energy level is $b_{i,min} + 1$. We need to discharge it first to energy level $b_{i,min}$ before it can be recharged. The discharging time is computed using Eq. (16), for $b_1 = b_{i,\mathcal{T}_{i,k}} = b_{i,min} + 1$ and $b_2 = b_{i,min}$, shown as the second term of the right hand side of Eq. (20). Then, we need to charge the battery from $b_{i,min}$ to $b_{i,max}$ before it can be used. We use Eq. (5) to compute the charging time interval, $\tilde{\tau}_i$. After fully charged, as per the HSU model, the battery can be discharged in the next slot, which explains the plus one in Eq. (20). □

Let α_i be the duration from time $\mathcal{T}_{i,k}$ to $t^+_{i,k+1}$, i.e., $\alpha_i = t^+_{i,k+1} - \mathcal{T}_{i,k}$; see Figure 6. Recall that $t^+_{i,k}$ is the starting time of a discharging cycle k of the battery at node v_i at which time the battery has energy level $b_{i,max} \times \mu_i$. The time duration α_i includes (i) time to discharge 1ϵ of energy, i.e., from $b_{i,min}+1$ to $b_{i,min}$ at cycle k before the battery can start charging at cycle $k+1$, (ii) time to charge the battery from level $b_{i,min}$ to $b_{i,max}$ at cycle $k+1$, and (iii) a one slot delay before the stored energy can be used at time $t^+_{i,k+1}$, as required by the HSU model. One can use Eq. (16) with $b_1 = b_{i,min} + 1$ and $b_2 = b_{i,min}$ to compute (i), and Eq. (5) to compute (ii). Thus, we have

$$\alpha_i = \lfloor \frac{log(b_{i,min}) - log(b_{i,min} + 1)}{log(\hat{\mu}_i)} \rfloor + \tilde{\tau}_i + 1. \tag{21}$$

Let β_i denote time duration from $t^+_{i,k}$ to $\mathcal{T}_{i,k}$, i.e., $\beta_i = \mathcal{T}_{i,k} - t^+_{i,k}$; see Figure 6. In other words, the battery takes β_i slots to discharge all its stored energy (due to leakage only) from level $b_{i,t^+_{i,k}} = b_{i,max} \times \hat{\mu}_i$ to $b_{i,min}+1$. Using Eq. (16), we have

$$\beta_i = \lfloor \frac{log\ (b_{i,min} + 1) - log\ (b_{i,max} \times \hat{\mu}_i)}{log\ \hat{\mu}_i} \rfloor. \tag{22}$$

The following proposition computes $t^+_{i,k+m}$ and $\mathcal{T}_{i,k+m}$ for the battery at node v_i in discharging cycle $k+m$, for integer $m = 0, 1, \cdots$.

Proposition 8: Consider the battery of node v_i in a discharging cycle k, and the battery is not used from time $t^+_{i,k}$ to $\mathcal{T}_{i,k+m}$, for any integer $m = 0, 1, \cdots$. For a given $\mathcal{T}_{i,k}$, the value of $t^+_{i,k+m}$ and $\mathcal{T}_{i,k+m}$ can be computed as

$$t^+_{i,k+m} = \mathcal{T}_{i,k} + m\alpha_i + (m-1)\beta_i, \tag{23}$$

$$\mathcal{T}_{i,k+m} = \mathcal{T}_{i,k} + m\alpha_i + m\beta_i. \tag{24}$$

Proof: For $m = 0$, we have $t^+_{i,k} = \mathcal{T}_{i,k} - \beta_i$, which is true by definition. For $m = 1$, we have $t^+_{i,k+1} = \mathcal{T}_{i,k} + \alpha_i$, which is also true by the definition of α_i or Eq. (21). To compute $t^+_{i,k+2}$, we must include the number of slots required to discharge the battery from the energy level at time $t^+_{i,k+1}$ to that at time $\mathcal{T}_{i,k+1}$, and the time duration from time $\mathcal{T}_{i,k+1}$ to time $t^+_{i,k+2}$. Thus, $t^+_{i,k+2} = \mathcal{T}_{i,k+1} + \alpha_i + \beta_i = \mathcal{T}_{i,k} + 2\alpha_i + \beta_i$. Similarly, for $m = 3$, we have $t^+_{i,k+3} = \mathcal{T}_{i,k+2} + \alpha_i + \beta_i = \mathcal{T}_{i,k} + 3\alpha_i + 2\beta_i$. Repeating the steps to compute $t^+_{i,k}$ and $\mathcal{T}_{i,k}$ for $m = 4, 5, \cdots$, we obtain Eq. (23) and Eq. (24), respectively. Note that for each pair of $\mathcal{T}_{i,k}$ and $t^+_{i,k}$ at any discharging cycle k, we have $\beta_i = \mathcal{T}_{i,k} - t^+_{i,k}$. □

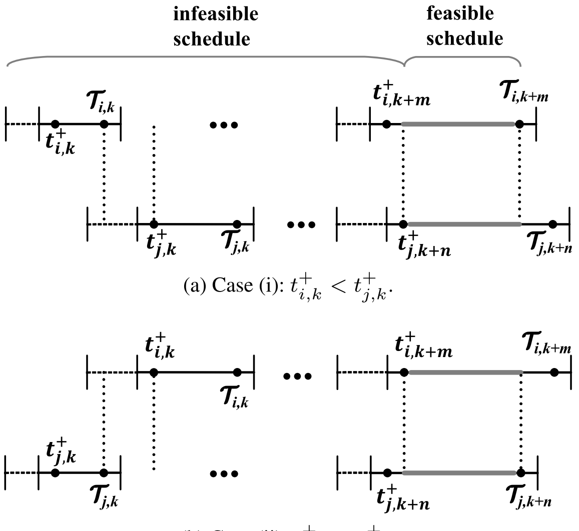
Figure 6 illustrates the relationship between α_i, β_i, $t^+_{i,k}$, and $\mathcal{T}_{i,k}$ used in Eq. (21), (22), (23), and (24), respectively. Note that variable $\mathcal{T}_{i,k}$ is used only for batteries with leakage rate $\mu_i > 0$. Thus, Proposition 6, 7, and 8 are not relevant for batteries with leakage rate $\mu_i = 0$.

We now describe a scenario to schedule a link $l_{i,j}$ in Figure 7. Consider link $l_{i,j}$ and case (i): $t^+_{i,k} < t^+_{j,k}$ in Figure 7a. Link $l_{i,j}$ can be activated at time slot $t_{i,j} = \mathbf{max}\{t^+_{i,k}, t^+_{j,k}\} = t^+_{j,k}$ only if (a) $t_{i,j} \geq t^+_{j,k+n}$, and (b) $t^+_{j,k+n} \leq \mathcal{T}_{i,k+m}$, for $m, n = 0, 1, \cdots$. Similarly, for case (ii): $t^+_{j,k} < t^+_{i,k}$ in Figure 7b, link $l_{i,j}$ can be activated at time slot $t_{i,j} = t^+_{i,k}$ only if (a) $t_{i,j} \geq t^+_{i,k+m}$, and (b) $t^+_{i,k+m} \leq \mathcal{T}_{j,k+n}$. In other words, we need to compute the two inequalities: (i) $t^+_{j,k+n} \leq \mathcal{T}_{i,k+m}$ and (ii) $t^+_{i,k+m} \leq \mathcal{T}_{j,k+n}$ to determine the time $t_{i,j}$ to schedule each link $l_{i,j}$. Using Eq. (23) and (24), for inequalities (i) and (ii), we obtain respectively

$$n(\alpha_j + \beta_j) - m(\alpha_i + \beta_i) \leq \mathcal{T}_{i,k} - \mathcal{T}_{j,k} + \beta_j, \tag{25}$$

$$m(\alpha_i + \beta_i) - n(\alpha_j + \beta_j) \leq \mathcal{T}_{j,k} - \mathcal{T}_{i,k} + \beta_i. \tag{26}$$

One can use SMT (satisfiability modulo theories) solver, e.g., Mistral [25], to solve linear inequalities over integers for expressions (25) and (26). The Mistral solver implements the Cuts-from-Proofs algorithm [26]. The solver is able to generate solutions for systems of linear inequalities that contain between 10 and 45 variables and between 15 and 50 inequalities per system. For problem **LSME**, it is preferable to find a pair (m, n) with the minimum of $\mathbf{max}\{m, n\}$ for inequalities (25) and (26) to minimize time $t_{i,j}$ so that link (i, j) can be scheduled earlier. Therefore, in this paper, we propose a simple heuristic function, called $find_{mn}(.)$, to obtain such a pair of (m, n) for inequalities (25) and (26). Function $find_{mn}$ $(i, j, \mathcal{T}_{i,k}, \mathcal{T}_{j,k})$ generates the values of m and n for inequalities (25) and (26) as follows. Initially it sets $(m, n) = (0, 0)$. From Eq. (23) and (24), increasing the value of m



(a) Case (i): $t^+_{i,k} < t^+_{j,k}$.

(b) Case (ii): $t^+_{j,k} < t^+_{i,k}$.

FIGURE 7. Two scenarios to schedule a link $l_{i,j}$.

results in a larger $t^+_{i,k+m}$ and $\mathcal{T}_{i,k+m}$ value. Similarly, a larger n value increases $t^+_{j,k+n}$ and $\mathcal{T}_{j,k+n}$. To minimize time $t_{i,j}$, $find_{mn}(.)$ needs to find the minimum integer values for m and n. Consider $t^+_{i,k} < t^+_{j,k}$, i.e., case (i) in Figure 7a, and $t^+_{j,k+n} > \mathcal{T}_{i,k+m}$. For this case, a larger value of $\mathcal{T}_{i,k+m}$ is needed. Thus, $find_{mn}(.)$ increases the value of m in (25) from $m = 0$ to $m' = 1$ to increase the value of $\mathcal{T}_{i,k+m}$, which also increases $t^+_{i,k+m}$. The function $find_{mn}(.)$ also finds a value of n' from (25) when it uses $m' = 1$. Next it produces a new value of m in (26) using n', e.g., m''. If $m'' \leq m'$, the values $(m' = 1, n')$ satisfy both (25) and (26), thus it returns (m', n'). However, if $m'' > m'$, it uses m'' in (25) to further increase the value of $\mathcal{T}_{i,k+m}$; it also obtains a new value for n, e.g., n''. If $n'' \leq n'$, it stops at a feasible solution (m'', n'). However, if $n'' > n'$, it uses n'' in (26) to generate an updated value of m, e.g., m''', which is then used to obtain another new value for n in (25), e.g., n'''. Function $find_{mn}(.)$ repeats the iterations until it finds the first feasible solution for m and n. Thus, link (v_i, v_j) can be scheduled at time $t_{i,j} = \mathbf{max}(t^+_{i,k+m}, t^+_{j,k+n})$.

To illustrate how the function $find_{mn}(.)$ works, consider link $l_{i,j}$ with $T_i = 22$, $\mathcal{T}_{i,k} = 74$, $T_j = 93$, $\mathcal{T}_{j,k} = 132$, $\alpha_i = 78$, $\alpha_j = 81$, $\beta_i = 90$, and $\beta_j = 39$. In this example, $T_j = t^+_{j,k}$ because $t^+_{j,k} = \mathcal{T}_{j,k} - \beta_j = 132 - 39 = 93$. However, we have $T_i \neq t^+_{i,k}$, meaning that the battery at node v_i has been used in discharging cycle k. From (25) and (26), we respectively have (i) $120n - 168m \leq -19$, and (ii) $168m - 120n \leq 148$. Notice that link $l_{i,j}$ cannot be activated at time $t_{i,j} = \mathbf{max}(22, 93) = 93$ because $t^+_{j,k} > \mathcal{T}_{i,k}$; see Figure 8. Thus, we need to *shift* the discharging cycle of battery at node v_i to the next cycle by increasing time $\mathcal{T}_{i,k+m}$. To shift one cycle, function $find_{mn}(.)$ sets $m = m' = 1$ in (i) and obtains $n' = 0$. As shown in Figure 8, the cycle now starts at $t^+_{i,k+1} = 152$ and ends at $\mathcal{T}_{i,k+1} = 242$, which are obtained by substituting $m = 1$ into Eq. (23) and (24), respectively. The function then sets $m = m' = 1$ in (ii) and gets $n' = 1 > 0$ that indicates the need for shifting the cycle for the battery at node v_j by one cycle. Otherwise, the link cannot be activated at time $t_{i,j} = \mathbf{max}(152, 93) = 152$ because $t^+_{i,k+1} > \mathcal{T}_{j,k}$; see Figure 8. Setting $n = n' = 1$ in (26), the function gets a new value of m, i.e., $m'' = 1$. Notice that $m'' \leq m'$, indicating that we don't need to further shift the duration for node v_i, i.e., $(m = 1, n = 1)$ is a feasible solution for (25) and (26). Thus, link $l_{i,j}$ can be scheduled at time $t_{i,j} = \mathbf{max}(t^+_{i,k+1}, t^+_{j,k+1}) = \mathbf{max}(152, 213) = 213$.

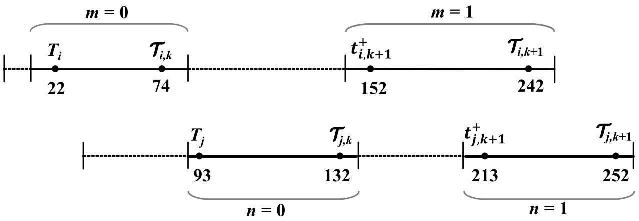


FIGURE 8. A discharging cycle of the battery at node v_i (top) and v_j (bottom).

B. PROBLEM SOLUTION FEASIBILITY

This section aims to show the feasibility of **LSME**. While in general two linear inequalities always have a solution, the values of m and n can be non integers [26]. One can use proof of unsatisfiability in [26] to determine if two linear inequalities have no integer solution. Thus, **LSME** for batteries with leakage rate $\mu_i > 0$ might not have a feasible solution, i.e., there can be some links $l_{i,j} \in E$ which cannot be scheduled. The reason is because link (i, j) can be scheduled only when the batteries at its end nodes are simultaneously in discharging cycle at the start of the same slot m or n, or different slots m and n. Recall that expressions (25) and (26) are for case $\mu_i > 0$.

Consider a link (i, j) and slot t at which time the batteries of end nodes v_i and v_j are in *charging* and *discharging* cycle, respectively. Thus, link (i, j) at time t cannot be scheduled. We call such link (i, j) an *unscheduled* link or u-link at time t, denoted by $(i, j)^t$, if its end nodes' other adjacent links will never be scheduled at or after time t. Let $E^t \subseteq E$ be a set of u-links at time t. Intuitively, when batteries of the end nodes of each u-link have the same recharging time, storage efficiency, leakage rate, minimum and maximum battery level, the batteries will never reach discharging cycle at the same time, and thus the link cannot be scheduled. The reason is because the batteries will have the same charging intervals as well as the same discharging intervals. The intuition is one possible necessary condition that prohibits expressions (25) and (26) having integers m and n as their solution. It is stated and formally proved in the following proposition.

Proposition 9: Each u-link $(i, j)^t \in E^t$ cannot be scheduled if all batteries have the same values of recharging time r_i, storage efficiency η_i, leakage rate μ_i, minimum battery level $b_{i,min}$, and maximum battery level $b_{i,max}$.

Proof: The proof considers three cases: (i) set E^t contains only one u-link $(i, j)^t$, (ii) set E^t contains more than one u-links where any pair of them are not adjacent to each other, and (iii) set E^t contains more than one u-links where some pair of them are adjacent to each other. For case (i), we show that u-link $(i, j)^t$ cannot be scheduled as follows. For $r_i = r_j$, $\mu_i = \mu_j$, $\eta_i = \eta_j$, $b_{i,min} = b_{j,min}$, $b_{i,max} = b_{j,max}$, Eq. (5) obtains $\tilde{\tau}_i = \tilde{\tau}_j$. Thus, for $\tilde{\tau}_i = \tilde{\tau}_j$, $b_{i,min} = b_{j,max}$, and $\mu_i = \mu_j$, Eq. (21) produces $\alpha_i = \alpha_j$. Similarly, Eq. (22) has $\beta_i = \beta_j$. Let α denote both α_i and α_j, and β denote both β_i and β_j. Further we set $\mathcal{T} = \mathcal{T}_{i,k} - \mathcal{T}_{j,k}$. Thus, we can convert expressions (25) and (26) into the following two expressions, respectively.

$$n(\alpha+\beta) - m(\alpha+\beta) \le \mathcal{T} + \beta, \tag{27}$$

$$m(\alpha+\beta) - n(\alpha+\beta) \le -\mathcal{T} + \beta. \tag{28}$$

One necessary condition for (27) and (28) to have integer solution for m and n is when $m \neq n$. Otherwise, the batteries at node v_i and node v_j cannot be simultaneously in discharging cycle because both batteries have the same aforementioned parameters, and thus they have the same charging interval and the same discharging interval. Now, we aim to show that any solution for expressions (27) and (28) cannot have integer values of m and n. Without loss of generality, consider $\mathcal{T}_{i,k} > \mathcal{T}_{j,k}$, and thus $\mathcal{T} > 0$. Multiplying both sides of (27) by -1, we have

$$m(\alpha+\beta) - n(\alpha+\beta) \ge -\mathcal{T} - \beta. \tag{29}$$

Thus, from (28) and (29), we have

$$-\mathcal{T} - \beta \le m(\alpha+\beta) - n(\alpha+\beta) \le -\mathcal{T} + \beta, \tag{30}$$

or

$$\frac{-(\mathcal{T}+\beta)}{(\alpha+\beta)} \le m - n \le \frac{(-\mathcal{T}+\beta)}{(\alpha+\beta)}. \tag{31}$$

Notice that we have $(-\mathcal{T}+\beta) < (\alpha+\beta)$ because $\mathcal{T}$ is a positive integer. Thus, we have $\frac{(-\mathcal{T}+\beta)}{(\alpha+\beta)} < 1$. Further, $\frac{-(\mathcal{T}+\beta)}{(\alpha+\beta)}$ has a negative value, or we have $\frac{-(\mathcal{T}+\beta)}{(\alpha+\beta)} < 0$. Thus, we have

$$\frac{-(\mathcal{T}+\beta)}{(\alpha+\beta)} < 0 \le m - n < \frac{(-\mathcal{T}+\beta)}{(\alpha+\beta)} < 1. \tag{32}$$

Since $m \neq n$, i.e., $m - n \neq 0$, we have

$$0 < m - n < 1, \tag{33}$$

which means the value of $m - n$ is a fraction. Thus, either variable m or n is a fraction.

For case (ii), we repeat the proof for case (i) for each u-link $(i, j)^t \in E^t$. Thus, all u-links in the set cannot be scheduled. Finally for case (iii), arbitrarily consider one u-link $(i, j)^t \in E^t$. Following the proof for case (i), u-link $(i, j)^t$ cannot be scheduled. Next, consider an adjacent link of $(i, j)^t$, e.g., a u-link $(i, k)^t$. We argue that $(i, k)^t$ cannot be scheduled because all of its parameters are unchanged. Note that the expressions (27) and (28) of u-link $(i, k)^t$ can have integer solutions m and n, and thus link (i, j) can be scheduled, only if the batteries' parameters, i.e., leakage rate μ_i, storage efficiency η_i, minimum battery level $b_{i,min}$, and maximum battery level $b_{i,max}$, change. □

To illustrate Proposition 9, consider the network in Figure 9 in which link $(1, 2)$ is activated first at slot 13. We have

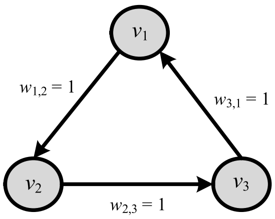


FIGURE 9. **A cycle with three nodes: an example for a non-existing solution. Each node v_i has $b_{i,min} = 1$, $b_{i,max} = 3$, $b_i = 3$, $\mu_i = 0.01$, $\eta_i = 1$, and $r_i = 5$.**

$T_1 = T_2 = 93$, $\mathcal{T}_{1,k} = \mathcal{T}_{2,k} = 132$, $T_3 = 13$, and $\mathcal{T}_{3,k} = 52$. Further, $E^t = \{(2, 3), (3, 1)\}$. We show that u-link $(2, 3)^t$ cannot be scheduled as follows. Eq. (5) obtains $\tilde{\tau}_2 = \tilde{\tau}_3 = 12$, while Eq. (21) and (22) produce $\alpha_2 = \alpha_3 = \alpha = 81$ and $\beta_2 = \beta_3 = \beta = 39$, respectively. Further, we have $\mathcal{T} = \mathcal{T}_{2,k} - \mathcal{T}_{3,k} = 132 - 52 = 80$. Thus, we have $-\frac{119}{120} < 0 \le m - n \le -\frac{41}{120} < 1$ for (32), and $0 < m - n < 1$ for (33), meaning either value of m or n is a fraction. Thus, u-link $(2, 3)^{t=93}$ cannot be scheduled and the parameters for u-link $(3, 1)^t$ remains unchanged. Similarly, u-link $(3, 1)^t$ cannot be scheduled because all of its parameters are same with u-link $(2, 3)^t$.

While it is important to find all necessary conditions that prohibit expressions (25) and (26) to have integer values of m and n for their solution, unfortunately, we fail to generate the conditions due to the complexity of the problem.

The following proposition formally shows that **LSME** for batteries with $\mu_i = 0$ always has a feasible solution.

*Proposition 10: **LSME** for batteries with leakage rate $\mu_i = 0$ will always have a feasible solution, i.e., it is possible to activate each link $l_{i,j} \in E$.*

Proof: For leakage rate $\mu_i = 0$, each battery at node v_i makes transition from discharging to charging cycle *only* after its energy is used to activate at least one incident link at v_i. Consider a link $l_{i,j} \in E$. Without loss of generality, assume the battery at node v_i has a longer discharging cycle than at node v_j. In other words, the battery at node v_i is less frequently used than at node v_j. For this case, the battery at node v_j will transition to charging cycle earlier than that at node v_i. Thus, the battery at node v_i remains in discharging cycle while the battery at node v_j is being recharged to its maximum level. When the battery at node v_j is back to discharging cycle, link $l_{i,j}$ can be activated. □

C. A METHOD TO SHORTEN SUPERFRAME LENGTH

The link schedule of an rWSN that complies with the battery cycle constraint will become longer when the batteries of nodes have a smaller leakage rate μ_i. The reason is as follows. Consider a battery of node v_i that has energy level of $b_{i,min} + x$, for $x < 1\epsilon$. The cycle constraint requires the energy level of the battery to reach $b_{i,min}$ before it can be recharged. However, discharging the non-usable energy from the battery gets longer when its leakage rate is smaller. As an example, consider a battery with $b_{i,min} = 1$ and $x = 0.5$. For leakage rate $\mu_i = 0.01$, $b_1 = 1.5$, and $b_2 = 1$, Eq. (16) produces 40 time slots to discharge the 0.5ϵ of energy.

The discharging time will significantly increase to 405464 slots when its leakage rate is $\mu_i = 10^{-6}$. The longer discharging time leads to longer schedule because the battery requires more time before it can be recharged to $b_{i,max}$ so that it can be used to activate links.

To shorten the discharging time, we assume each node is able to discard or *flush* its excess energy x at slot t in the next slot $t+1$. One way for a node to *flush* excess energy of size x is by using the energy to transmit a *dummy* packet in a given slot. This protocol, henceforth called *energy flush*, is similar to that assumed in [20]. Briefly, the work in [20] uses two batteries, called primary and secondary, which are respectively in discharging and charging mode. When the secondary battery is fully charged, it becomes the primary battery, while the other becomes the secondary battery, i.e., in charging mode, even when its energy level has not reached the minimum, i.e., $b_{i,min}$ in our model.

The use of the *energy flush* protocol affects the computation of Eq. (16) when it has $b_1 = b_{i,min} + x$ for $x < 1$, and $b_2 = b_{i,min}$. For this case, the value of $\delta_{i,b_1}^{b_2}$ is set to one because according to the energy flush protocol the x excess energy can be discharged in one slot. Consequently, for this case, we need to set the value of $\delta_{i,b_1}^{b_2}$ in Eq. (18) to one. Thus, Eq. (18) for $\mu_i > 0, b_1 = b_{i,min}+x$ for $x < 1$, and $b_2 = b_{i,min}$ becomes

$$T_i = t_i + \sigma_{i,t_i} \times (1 + \tilde{\tau}_i) + 1 \qquad (34)$$

D. ALGORITHM

This section provides the details of our heuristic algorithm to solve **LSME**, called Link Scheduling with Battery Cycle Constraint (LSBCC). The algorithm aims to schedule all non-interfering links at the earliest possible time slot when the battery at its end nodes can be used, i.e., in discharging mode. Further, it uses the conflict graph C_G to check the interference links.

Algorithm 1 LSBCC

Input: $G(V, E)$, r_i, b_i, $b_{i,max}$, $b_{i,min}$, μ_i, η_i of each node $v_i \in V$, weight $w_{i,j}$ of each link $l_{i,j} \in E$, and conflict graph C_G

Output: Superframe $\mathcal{S}$

```
1:  for each node $v_i \in V$ do
2:      $INIT(\tilde{b}_{i,0}, \tilde{\tau}_i, \alpha_i, \beta_i, t_i, T_i, \mathcal{T}_{i,k}, b_{i,T_i})$
3:  end for
4:  for each link $l_{i,j} \in E$ do
5:      $t_{i,j} = COMP\_t_{i,j}(v_i, v_j)$
6:  end for
7:  $K$ = {node $l_{i,j}$ in $C_G$ with $min\{t_{i,j}\}$}
8:  $K' = ORDER(K)$
9:  $t \leftarrow \mathbf{min}\{t_{i,j}\}$
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:         $b_{i,t_i} \leftarrow b_{i,T_i} \times \hat{\mu}_i^{t_i - T_i} - 1$
19:         $b_{j,t_j} \leftarrow b_{j,T_j} \times \hat{\mu}_j^{t_i - T_j} - 1$
20:         $T_i \leftarrow COMP\_T_\alpha(i)$
21:         $b_{i,T_i} \leftarrow COMP\_b_\alpha(i)$
22:         $T_j \leftarrow COMP\_T_\alpha(j)$
23:         $b_{j,T_j} \leftarrow COMP\_b_\alpha(j)$
24:         $\mathcal{T}_{i,k} \leftarrow T_i + \lfloor \frac{log\ (b_{i,min}+1) - log\ b_{i,T_i}}{log\ \hat{\mu}_i} \rfloor$
25:         $\mathcal{T}_{j,k} \leftarrow T_j + \lfloor \frac{log\ (b_{j,min}+1) - log\ b_{j,T_j}}{log\ \hat{\mu}_j} \rfloor$
26:     end if
27: end for
28: repeat Line 4-27 until all $w_{i,j} = 0$
```

We first describe LSBCC in Algorithm 1 that considers batteries with leakage rate $\mu_i > 0$. LSBCC sets time $t = 0$ as the beginning of time slot and battery of each node is initially in charging mode. In *Lines 1-3*, LSBCC uses function *INIT*(.) to initialize the following eight parameters for each node v_i. The function initializes energy level of battery at each node v_i to $b_{i,min}$, i.e., $\tilde{b}_{i,0} = b_{i,min}$. It uses Eq. (5) to compute $\tilde{\tau}_i$, i.e., the charging time interval for the battery at node v_i. It sets time duration α_i and β_i using Eq. (21) and (22), respectively. It initializes t_i to the last time the battery at node v_i is discharged to zero. It also sets T_i to $\tilde{\tau}_i + 1$ and $\mathcal{T}_{i,k}$ to $T_i + \beta_i$. Recall that T_i and $\mathcal{T}_{i,k}$ are the earliest time and the latest time slot when the battery at node v_i can be discharged to transmit or receive one packet. Finally, it initializes b_{i,T_i}, the energy level of battery at node v_i at time T_i, to $b_{i,max} \times \hat{\mu}_i$, i.e., $b_{i,T_i} = b_{i,max} \times \hat{\mu}_i$.

Lines 4-6 use function $COMP\_t_{i,j}(.)$ in Algorithm 2 to compute the earliest time each link (i,j) can be activated, i.e., $t_{i,j}$. If the batteries of node v_i and v_j are at the same discharging cycle, *Line 2* of $COMP\_t_{i,j}(.)$ sets $t_{i,j}$ to the earliest time at which both batteries can be discharged. On the other hand, when the two batteries are not at the same discharging cycle, *Lines 4* to *15* aim to obtain the earliest time that the battery at nodes v_i and v_j can be at the same discharging cycle. More specifically, *Line 4* first uses the proof of unsatisfiability [26] in function *SATISFY*(.) to determine if (25) and (26) for link (i,j) have integer solutions. If so, *Line 5* uses function $find_{mn}(.)$ to compute a pair (m,n) that satisfies expressions (25) and (26). Otherwise, *Line 14* sets $t_{i,j}$ to a large integer value, i.e., $2^{31}-1$, to denote that link $l_{i,j}$ can not be scheduled in this iteration. Note that, it is possible that link $l_{i,j}$ can be scheduled in the future. *Lines 6-7* compute the starting time of the next discharging cycle of the battery at node v_i and v_j. *Lines 8-9* obtain the battery level of node v_i and v_j. *Lines 10-11* recompute the ending time of the next discharging cycle of the battery at node v_i and v_j. *Line 12* computes $t_{i,j}$.

Line 7 of LSBCC creates a set K that stores all links (i,j) that have the same earliest activation time. *Line 8* then uses function $ORDER(K)$ to sort links in set K in order of

Algorithm 2 *COMP_$t_{i,j}$*

```
Input: $v_i$, $v_j$
Output: $t_{i,j}$
 1: if $T_j \le \mathcal{T}_{i,k}$ or $T_i \le \mathcal{T}_{j,k}$ then
 2:     $t_{i,j} \leftarrow \mathbf{max}(T_i, T_j)$
 3: else
 4:     if SATISFY$(i, j)$ is true then
 5:         $find_{mn}(i, j, \mathcal{T}_{i,k}, \mathcal{T}_{j,k})$
 6:         $t^+_{i,k+m} \leftarrow \mathcal{T}_{i,k} + m\alpha_i + (m-1)\beta_i$
 7:         $t^+_{j,k+n} \leftarrow \mathcal{T}_{j,k} + n\alpha_j + (n-1)\beta_j$
 8:         $b_{i,t^+_{i,k+m}} \leftarrow b_{i,max} \times \hat{\mu}_i$
 9:         $b_{j,t^+_{j,k+n}} \leftarrow b_{j,max} \times \hat{\mu}_j$
10:         $\mathcal{T}_{i,k+m} \leftarrow \mathcal{T}_{i,k} + m\alpha_i + m\beta_i$
11:         $\mathcal{T}_{j,k+n} \leftarrow \mathcal{T}_{j,k} + n\alpha_i + n\beta_i$
12:         $t_{i,j} \leftarrow \mathbf{max}(t^+_{i,k+m}, t^+_{j,k+n})$
13:     else
14:         $t_{i,j} \leftarrow 2^{31} - 1$
15:     end if
16: end if
17: return$(t_{i,j})$
```

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., $\mathbf{min}\{t_{i,j}\}$. *Lines 10-27* 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 the conflict graph of the network; i.e., $M[a, b]$ is set to "1" if there is interference between links a and b. *Line 13* decreases the weight of each selected link $l_{i,j}$ by one. Once the weight reaches zero (see *Line 14*), *Line 15* removes the link from contention.

Line *17* sets the last time batteries at the end nodes of selected link $l_{i,j}$ is used, i.e., t_i and t_j, to the current time. Further, *Lines 18-19* compute the energy level of the battery at node v_i and v_j at time t_i after being used to transmit/receive one packet. *Line 20* uses function *COMP_T_α*(.) that implements Eq. (18) to update the next earliest time the battery at node v_i can be discharged. *Line 21* uses function *COMP_b_α*(.) to compute the battery's energy level at time T_i, which depends on the remaining energy level at time t_i, i.e., b_{i,t_i}. More specifically, if b_{i,t_i} is larger or equal to $(b_{i,min} + 1)$, then b_{i,T_i} is set to $b_{i,t_i} \times \hat{\mu}_i$ as the battery still can be used to transmit/receive one packet. Otherwise, b_{i,T_i} is set to $b_{i,max} \times \hat{\mu}_i$ since the battery has been charged to its maximum level. Similarly, LSBCC computes the next earliest time and energy level of the battery at node v_j in *Lines 22-23*. *Lines 24-25* use Eq. (19) to obtain the latest time the battery at node v_i and v_j can be used. Finally, the steps from *Line 4* is repeated until all links is scheduled, i.e., $w_{i,j} = 0$.

Next, we describe how to adjust LSBCC in Algorithm 1 for use in the case when each battery at node v_i is leak-free, i.e., for case $\mu_i = 0$. The adjustment comprises the following four changes to Algorithm 1: a) Function *INIT*(.) in *Line 2* of LSBCC does not initialize parameters α_i, β_i, and $\mathcal{T}_{i,k}$. Recall that these three parameters are used only for batteries with $\mu_i > 0$; b) Replace *Line 5* of Algorithm 1 with $t_{i,j} = \mathbf{max}(T_i, T_j)$. Note that function *COMP_$t_{i,j}$*(.) is applicable only for batteries with $\mu_i > 0$; c) Set b_{i,T_i} to $b_{i,max}$ in *Line 21* if b_{i,t_i} is equal to $b_{i,min}$. Otherwise, set b_{i,T_i} to b_{i,t_i}. Do similar adjustment in *Line 23* as in *Line 21*; and lastly, d) Omit *Lines 24-25*.

For an example, consider the rWSN and conflict graph C_G shown in Figure 3. The function *INIT*(.), for each node v_i, sets the following eight parameters as: (i) $\tilde{b}_{1,0} = \tilde{b}_{2,0} = \tilde{b}_{3,0} = \tilde{b}_{4,0} = b_{1,min} = 1$; (ii) $\tilde{\tau}_1 = 9$, $\tilde{\tau}_2 = 12$, $\tilde{\tau}_3 = 20$, $\tilde{\tau}_4 = 10$; (iii) $\alpha_1 = 78$, $\alpha_2 = 81$, $\alpha_3 = 89$, $\alpha_4 = 79$; (iv) $\beta_1 = 90$, $\beta_2 = \beta_3 = 39$, $\beta_4 = 67$; (v) $t_1 = t_2 = t_3 = t_4 = 0$; (vi) $T_1 = \tilde{\tau}_1 + 1 = 10$, $T_2 = 13$, $T_3 = 21$, $T_4 = 11$; (vii) $\mathcal{T}_{1,k} = T_1 + \beta_1 = 100$, $\mathcal{T}_{2,k} = 52$, $\mathcal{T}_{3,k} = 60$, $\mathcal{T}_{4,k} = 78$; and (viii) $b_{1,T_1} = b_{1,max} \times \hat{\mu}_1 = 5 \times 0.99 = 4.95$, $b_{2,T_2} = 2.97$, $b_{3,T_3} = 2.97$, $b_{4,T_4} = 3.96$. *Lines 4-6* obtain $t_{1,2} = 13$, $t_{2,4} = 13$, and $t_{3,1} = 21$. *Line 7* inserts links $l_{1,2}$ and $l_{2,4}$ into the set K, and thus *Line 8* obtains $K' = \{l_{2,4}, l_{1,2}\}$ because $w_{2,4} > w_{1,2}$, and *Line 9* sets $t = 13$. *Line 11* finds that $l_{2,4}$ has conflict with $l_{1,2}$, and thus *Line 12* inserts only link $l_{2,4}$ into $\mathcal{S}[13]$, and *Line 13* reduces $w_{2,4}$ by one and hence it becomes two. *Line 17* sets $t_2 = t_4 = 13$. *Lines 18-19* compute $b_{2,13} = 1.97$ and $b_{4,13} = 2.88$. *Lines 20-23* obtain $T_2 = 93$, $b_{2,T_2} = 2.97$ and $T_4 = 14$, $b_{4,T_4} = 2.85$. *Lines 24-25* then compute $\mathcal{T}_{2,k} = 132$ and $\mathcal{T}_{4,k} = 49$. *Line 28* repeats the steps from *Line 4* until all links have $w_{i,j} = 0$. Finally, LSBCC produces the link schedule $\mathcal{S}$ in Figure 4b, i.e., $\mathcal{S} = [\,\mathcal{S}[13] = \{l_{2,4}\}, \mathcal{S}[21] = \{l_{3,1}\}, \mathcal{S}[128] = \{l_{2,4}\}, \mathcal{S}[152] = \{l_{2,4}\}, \mathcal{S}[227] = \{l_{1,2}\}, \mathcal{S}[307] = \{l_{1,2}\}\,]$. The generated schedule contains 303 empty slots as the battery at each node needs time to charge to its maximum level before it can be used to transmit/receive packets.

Proposition 11: The time complexity of LSBCC is $O(W|E|^2)$, where $W = \sum_{(i,j)\in|E|} (w_{i,j})$.

Proof: *Lines 1-3* take $O(|V|)$. *Lines 4-6* require $O(|E|)$ because *Lines 1-17* of function *COMP_$t_{i,j}$* take $O(1)$ each and these lines are repeated at most $|E|$ times. *Line 7* takes $O(|E|)$. *Line 8* sorts all links in K using the function *ORDER*(K) that requires $O(|E|\ log\ |E|)$. *Line 9* takes $O(1)$. *Line 11* requires $O(|E|^2)$ to construct a matrix M which represents the conflict graph C_G. Function *CONFLICT*$(l_{i,j}, \mathcal{S}[t])$ in *Line 11* uses the matrix at most $|E|$ times. Hence, it takes $O(|E|)$. *Lines 12-25* take $O(1)$ each. The for loop in *Lines 10-27* is repeated at most $|E|$ times, and thus, the loop requires at most $O(|E|^2)$. *Line 28* repeats *Lines 4-27* W times. Thus, the time complexity of LSBCC is $O(W|E|^2)$. □

Notice that the time complexity of LSBCC becomes $O(|E|^3)$ if each link weight has a constant value. For this case, LSBCC runs in polynomial order of the number of links $|E|$.

TABLE 2. Parameter values used in our evaluation.

Parameter	*Value (s)*
Network size	40 x 40 m^2
Transmit range	15 m
Interference range	30 m
$\lvert V\rvert$	$\{10, 20, 30, 40, 50\}$
r_i	$\{2, 3, 4, \cdots, 17\}$
μ_i	$\{1, 1.2, 1.4, 1.6, \cdots, 2.4\} \times 10^{-6}$
η_i	1.0
b_i	3ϵ
$b_{i,min}$	1ϵ
$b_{i,max}$	3ϵ
$w_{i,j}$	$\{1, 2, 3, 4, 5\}$

Further, the running time of the algorithm worsens on networks that contain a high number of links. Nevertheless, since LSBCC runs in polynomial order of $|E|$, it is scalable for use in larger sized networks.

IV. EVALUATION

We have implemented LSBCC in C++ and conducted our experiments on a computer with an Intel Core i7 CPU @ 3.4 GHz and 16 GB of RAM. Section IV-A analyzes the schedule length when nodes use a battery that adheres to the battery cycle constraint but with *no leakage*. Section IV-B aims to analyze the effects of parameters r_i, b_i, $b_{i,max}$, and η_i on the feasibility of **LSME** when nodes have a battery with leakage rate $\mu_i > 0$. Finally, Section IV-C evaluates the impact of the battery cycle constraint on link schedule length and the number of charge/discharge cycles for rWSNs with a leak-free battery and one that leaks.

Table 2 lists the parameter values used in our simulation. We consider arbitrary networks with 10 to 50 nodes randomly deployed on a 40 × 40 m^2 area. The average number of links $|E|$ is 28, 125, 273, 470, and 758 for networks with 10, 20, 30, 40, and 50 nodes respectively. Each node has a transmit and interference range of 15 and 30 meters, respectively, i.e., as in [14] the interference range is two times the transmit range. We arbitrarily set the range of values of r_i and $w_{i,j}$ to $\{2, 3, \ldots, 17\}$ and $\{1, 2, \ldots, 5\}$, respectively. We use leakage rate μ_i values in the set $\{1, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4\} \times 10^{-6}$ per slot; also, the battery leakage rate is 20% per 24 hours [27]. We set each slot to one second. Further, we consider batteries with 100% storage efficiency, and arbitrarily set the values of b_i, $b_{i,min}$, and $b_{i,max}$. Note that as reported in [15], battery capacity has insignificant effect on superframe length $|\mathcal{S}|$. Our results are an average over 100 random node deployments.

A. BATTERY WITH NO LEAKAGE

The aim of this section is to study how the battery cycle constraint affects the superframe length $|\mathcal{S}|$ when each battery has no leakage, i.e., $\mu_i = 0$. Recall that, as stated in Proposition 10, **LSME** always has a feasible solution for $\mu_i = 0$. This section performs two evaluations. First, it compares the performance of LSBCC against LSNBC. Briefly, LSNBC is a version of LSBCC with no battery cycle constraint. Second, it investigates the effect of energy harvesting time r_i on the link schedules produced by LSBCC. All experiments consider 10 to 50 nodes with the following parameter values: $b_i = 3\epsilon$, $b_{i,max} = 3\epsilon$, $b_{i,min} = 1\epsilon$.

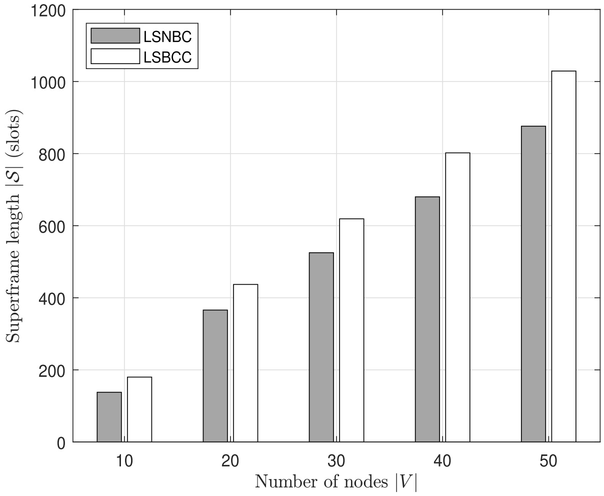


FIGURE 10. LSBCC versus LSNBC in terms of superframe length $|\mathcal{S}|$.

1) LSBCC VERSUS LSNBC

Each link weight $w_{i,j}$ is drawn randomly from $[1, 5]$, and $r_i = 5$. Figure 10 shows that the superframe length $|\mathcal{S}|$ produced by LSBCC is longer than LSNBC. In a rWSN with 10 nodes, LSBCC produces 42 more slots (30.43% longer) as compared to when using LSNBC. The results are consistent for other networks, i.e., $|V| = 20, 30, 40, 50$ nodes. LSBCC produces superframes that are 19.4%, 17.9%, 17.94%, and 17.47% longer than in LSNBC, respectively, with standard deviation values ranging between 56 and 129. The reason is because nodes using LSBCC need to wait for their battery to be fully charged before they can discharge their battery.

2) EFFECT OF HARVESTING TIME

In this simulation, we consider various r_i values, namely $1, 5, 10, 15, 20$ slots. Links have a weight of $w_{i,j} = 3$. From Figure 11, we see that energy harvesting time of nodes has a significant effect on $|\mathcal{S}|$, i.e., increasing their energy harvesting time results in a longer superframe. Specifically, for a rWSN with 10 nodes, when r_i is increased by four slots, i.e., from one to five, $|\mathcal{S}|$ jumps from 68 to 173 slots - an increase of 1.54 times. Similarly, when r_i increases from 5 to 20 with an interval of five, i.e., from five to 10, 10 to 15, and 15 to 20, $|\mathcal{S}|$ is further increased by 139, 140, and 140 slots, meaning the link schedule increases by 0.8, 0.45, and 0.31 times, respectively. We observe similar trends in rWSN with 20, 30, 40, and 50 nodes. For example, for a rWSN with 50(100) nodes, when r_i increases from 1 to 20 with an interval of five, $|\mathcal{S}|$ is increased by 1.33(1.34), 0.72(0.72), 0.42(0.42), 0.3(0.3) times respectively. The increase in $|\mathcal{S}|$ is because each battery needs more time to be charged to its maximum level before it can be used to transmit/receive a packet. Also notice that the $|\mathcal{S}|$ for each network size increases almost linearly when r_i is increased from one to 20. Further, the rate of increase (in slots) in smaller networks, e.g., $|V| = 10$, is less than that of larger networks, e.g., $|V| = 50$. The reason

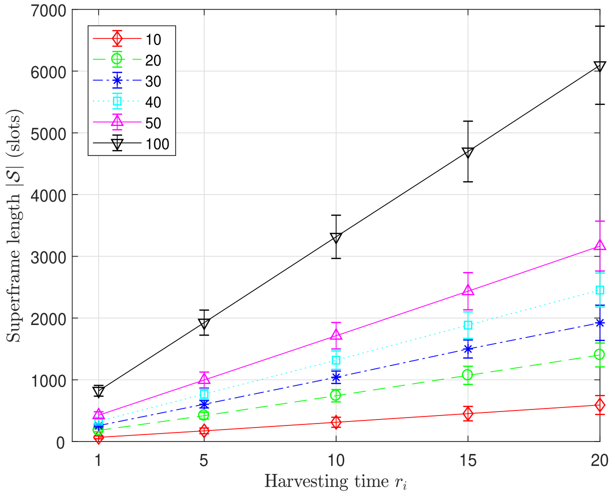


FIGURE 11. Effect of harvesting time r_i on the superframe length $|\mathcal{S}|$.

TABLE 3. The Number of Scheduled Links (In %).

Various Cases		|V|				
		10	20	30	40	50
Case (1)	$w_{i,j} = 1$	21.43	7.2	6.59	4.47	3.56
	$w_{i,j} = 3$	8.33	2.4	2.08	1.56	1.19
Case (2)	$w_{i,j} = 1$	78.57	91.20	94.87	97.59	98.68
	$w_{i,j} = 3$	84.52	89.07	94.26	96.81	97.63
Case (3)	$w_{i,j} = 1$	100	100	100	100	100
	$w_{i,j} = 3$	100	100	100	100	100
Case (4)	$w_{i,j} = 1$	100	100	100	100	100
	$w_{i,j} = 3$	100	100	100	100	100
Case (5)	$w_{i,j} = 1$	100	100	99.63	95.53	90.24
	$w_{i,j} = 3$	97.62	100	99.15	95.53	93.18

is because more nodes mean more links need to be scheduled. Also, more links will have to wait for sufficient energy before they can be activated. Figure 11 also shows that the increase in $|\mathcal{S}|$ is more significant in denser networks; see the results for 100 nodes.

B. FEASIBLE SOLUTIONS OF LSME

This experiment aims to empirically demonstrate that the **LSME** in general does not always have a feasible solution, i.e., some links cannot be scheduled for some battery leakage rate $\mu_i > 0$. More specifically, it analyzes the impact of parameters r_i, b_i, $b_{i,max}$, η_i, and $\mu_i > 0$ on **LSME**'s solution. The experiment considers five cases, i.e., all nodes have 1) the same parameter values, 2) random parameter values, 3) different values of μ_i, 4) different values of r_i, and 5) different values of pair (r_i, μ_i). Each case considers two different values of link weight, i.e., $w_{i,j} = 1$ and $w_{i,j} = 3$. Table 3 summarizes the number of scheduled links for each case. The proof of unsatisfiability [26] is used on inequalities (25) and (26) of each link (i, j) to determine if the link can be scheduled.

1) SAME PARAMETER VALUES

All nodes use the same values of parameters r_i, b_i, $b_{i,max}$, μ_i, and η_i, each of which is selected randomly from set $\{2, 5\}$, $\{3, 4\}$, $\{3, 4\}$, $\{1, 2.2\} \times 10^{-6}$, and $\{0.9, 1\}$, respectively With weight $w_{i,j} = 1$, most links cannot be scheduled, i.e., only 21.43% (6 of 28), 7.2% (9 of 125), 6.59% (18 of 273), 4.47% (21 of 470), and 3.56% (27 of 758) of the total number of links in networks with 10, 20, 30, 40, and 50 nodes respectively, can be scheduled; see Case (1) in Table 3. Recall that a link (i, j) can be scheduled only if the batteries of its end nodes i and j have an energy level at least one ϵ **and** are in the same *discharging* cycle. Alternatively, link (i, j) cannot be scheduled because one fails to find integer values of m and n that satisfy the expressions 25 and (26). Similar results are obtained for $w_{i,j} = 3$, where each link needs to be scheduled three times. More specifically, only 8.33% (7 of $3 \times 28 = 84$), 2.4% (9 of 375), 2.08%, 1.56%, and 1.19% of the total link schedules can be generated for networks with 10, 20, 30, 40, and 50 nodes, respectively.

2) RANDOM PARAMETER VALUES

Each node is assigned with randomly generated parameters. More specifically, we randomly set $r_i \in \{2, 3, 4, 5\}$, $b_i \in \{3, 4\}$, $b_{i,max} \in \{3, 4\}$, $\mu_i \in \{1, 1.4, 1.8, 2.2\} \times 10^{-6}$, and $\eta_i \in \{0.9, 1\}$. The average number of scheduled links increases when we assign random parameter values at nodes. As an example, Table 3 shows that for $w_{i,j} = 3$ and $|V| = 30$, Case (2) produces only 5.74% unscheduled links as compared to 97.92% in Case (1). Notice that the results for $w_{i,j} = 1$ and $w_{i,j} = 3$ for Case (2) are consistent. As an example, for $|V| = 50$ and $w_{i,j} = 1$ ($w_{i,j} = 3$), LSBCC is able to schedule 98.68% (97.63%), i.e., 740 of 758 (2244 of 2274) links.

3) DIFFERENT VALUES OF μ_i

Each node is set to have the same value of parameter $r_i = 2$, $b_i = 3\epsilon$, $b_{i,min} = 1\epsilon$, $b_{i,max} = 3\epsilon$, $\eta_i = 1$. Different values of μ_i are assigned to the end nodes of each link as follows: (i) use the *chromatic number* algorithm [28] to compute the minimum number of different values of μ_i for each network; (ii) use the *vertex coloring* algorithm [28] to assign the end nodes of each link with different values of $\mu_i \in \{1, 1.2, 1.4, 1.6, \cdots, 2.4\} \times 10^{-6}$.

As shown in Case (3) of Table 3, the average number of scheduled links reaches 100% for all network sizes. However, it does not guarantee the feasibility of **LSME**'s link schedule. For example, when $r_i \in \{2, 4\}$ and with other parameter retaining the same value, 99.56% of links can be scheduled for $|V| = 50$ with $w_{i,j} = 3$. Note that, this result is not shown in Table 3 to reduce space.

4) DIFFERENT VALUES OF r_i

Each node is assigned with the same value of the following parameters: $b_i = 3\epsilon$, $b_{i,min} = 1\epsilon$, $b_{i,max} = 3\epsilon$, $\mu_i = 1 \times 10^{-6}$, and $\eta_i = 1$. However, the end nodes of each link use different value of harvesting time r_i, each of which is randomly drawn from set $\{2, 3, 4, \cdots, 17\}$. As in Section IV-B3, the chromatic number and vertex coloring algorithms are used to assign different values of r_i to the end nodes of each link. For link weight $w_{i,j} = 1$ and $w_{i,j} = 3$, the total number of links that can be scheduled are 100% for $|V| = 10, 20, 30, 40, 50$ nodes; see Case (4) in Table 3. However, setting the end nodes of each link with a different harvesting time increases the number of scheduled links.

5) DIFFERENT VALUES OF PAIR (r_i, μ_i)

Each node is assigned with the same value of the following parameters: $b_i = 3\epsilon$, $b_{i,min} = 1\epsilon$, $b_{i,max} = 3\epsilon$, and $\eta_i = 1$. However, the end nodes of each link have different values of pair (r_i, μ_i), for harvesting time $r_i \in \{2, 3, 4, 5, 6\}$ and leakage rate $\mu_i \in \{1, 1.2, 1.4, 1.6, \cdots, 2.4\} \times 10^{-6}$. Hence, there are $5 \times 8 = 40$ different (r_i, μ_i) pairs, i.e., $(2, 1 \times 10^{-6})$, $(2, 1.2 \times 10^{-6})$, $(2, 1.4 \times 10^{-6})$, $(2, 1.6 \times 10^{-6})$, $\cdots$, $(2, 2.4 \times 10^{-6})$, $(3, 1 \times 10^{-6})$, $\cdots$, $(6, 2.4 \times 10^{-6})$. As in Section IV-B3, the chromatic number and vertex coloring algorithms [28] are used to assign different pairs of (r_i, μ_i) to the end nodes of each link. As shown in Case (5) of Table 3, the average number of links which can be scheduled is 100%, only for networks with 10 ($w_{i,j} = 1$) and 20 nodes. Table 3 shows that, for $w_{i,j} = 1$ ($w_{i,j} = 3$), LSBCC is able to schedule 99.63% (99.15%), 95.53% (95.53%), and 90.24% (93.18%) of the links in networks with 30, 40, and 50 nodes, respectively.

The results in Table 3 show that, except for Case (2), the percentage of links that can be scheduled decreases for networks with larger number of nodes. The reason is because larger networks have more links and hence have a higher probability that links have end nodes that do not have the same discharging cycle. Thus, these links cannot be scheduled. Further, assigning the end nodes of each link with different values of μ_i, r_i, or pair (r_i, μ_i), i.e., in Case (3), (4), and (5) respectively, tend to reduce the number of unscheduled links.

C. LEAK-FREE VERSUS LEAK BATTERY

This section aims to compare the impact of the battery cycle constraint on networks where nodes have a leak-free battery and those with a battery that leaks. Firstly, it presents the results of a simulation that evaluates the effect of battery cycle constraint on the link schedule. The results compare the schedule length $|\mathcal{S}|$ generated by LSBCC against LSNBC. Secondly, it aims to see which constraint, *charging* or *discharging*, has a larger effect on the schedule length. Recall that a battery charging constraint enforces the battery to be charged only when its capacity reaches the minimum level. On the other hand, a battery discharging constraint imposes the battery can be used only when its energy level reaches the maximum level. Here we compare the schedule length generated by two versions of LSBCC: LSCC that considers only charging constraint, and LSDC that enforces only discharging constraint. Finally, it provides the results of a simulation that examines the effect of the battery cycle constraint on the number of charge/discharge cycles. Each simulation considers 10 to 50 nodes and uses the following parameter values: $b_i = 3\epsilon$, $b_{i,max} = 3\epsilon$, $b_{i,min} = 1\epsilon$, and $\eta_i = 1$. Note that for all simulations, the parameters r_i and μ_i are set to values that ensure there is a feasible link schedule.

1) LSBCC VERSUS LSNBC

In this simulation, we set $r_i \in \{2, 3, 4, 5, 6\}$ and $w_{i,j} = 3$. The leakage rate μ_i of batteries is set to a value in $\{1, 1.2, 1.4, 1.6, 1.8\} \times 10^{-6}$. We first evaluate the effect of using the energy flush method, discussed in Section III-C, on superframe length $|\mathcal{S}|$. Let LSBCC^f (LSBCC^{nf}) denote LSBCC that does (does not) use energy flush. Then we compare the schedule length produced by LSBCC^f against that by LSNBC.

TABLE 4. Superframe length $|\mathcal{S}|$ of LSBCC^f and LSBCC^{nf}.

$\lvert V\rvert$	LSBCC^f	LSBCC^{nf}
10	327	13336041
20	841	38888681
30	1247	54636058
40	1618	116185914
50	2106	149635876

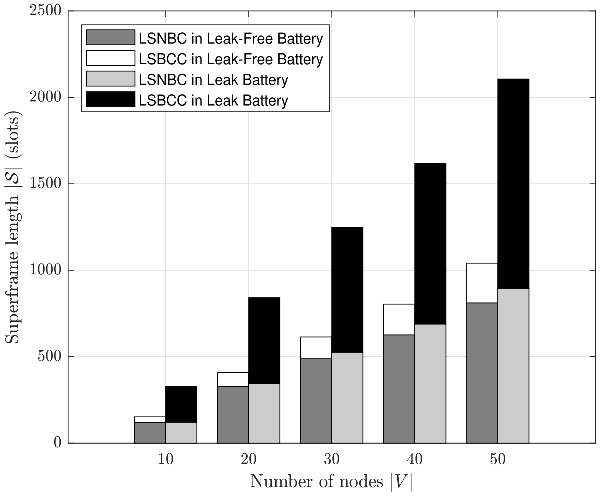


FIGURE 12. LSBCC versus LSNBC in networks where nodes have a leak-free and leak battery in terms of superframe length $|\mathcal{S}|$.

As shown in Table 4, all superframes generated by LSBCC^{nf} are significantly longer than LSBCC^f. For example, in networks with $|V| = 10$ nodes, LSBCC^{nf} produces 13335714 more slots (40782 times longer) as compared to those generated by LSBCC^f. Similarly, LSBCC^{nf} generates 46240, 43813, 71807, and 71051 times longer superframes than those by LSBCC^f in networks with $|V| = 20, 30, 40, 50$ nodes, respectively. As explained in Section III-C, discharging a battery when its energy level is less than $b_{i,min} + 1$ takes vast number of slots when its leakage rate is very small, e.g., 10^{-6}.

In the remaining simulations, we use only LSBCC that utilizes the energy flush method.

Figure 12 shows that the superframe length $|\mathcal{S}|$ produced by LSBCC is longer than LSNBC. As an example, for networks where nodes have a leak-free battery and $|V| = 10$ nodes, LSBCC produces 33 more slots (27.73% longer) as compared to when using LSNBC. The results are consistent for other networks, i.e., $|V| = 20, 30, 40, 50$ nodes. LSBCC produces superframes that are 24.77%, 25.82%, 28.43%, and 28.36% longer than in LSNBC, respectively, with standard deviation values ranging between 63 and 143. For rWSNs with a battery that leaks, LSBCC generates 206 more slots (1.7 times longer) than LSNBC in $|V| = 10$. Similarly, LSBCC produces 1.43, 1.37, 1.35, and 1.35 times longer superframes than LSNBC when there are

$|V| = 20, 30, 40, 50$ nodes, respectively, with standard deviation values ranging between 61 and 298. The reason is because LSBCC needs to wait for the battery at each node to be charged (discharged) to its maximum (minimum) level before it can be discharged (charged). These results show that the battery cycle constraint of each battery result in a negative impact on the schedule length. Note that, this experiment produces similar results for link weight $w_{i,j} = 1$; the results are not presented to save space.

Figure 12 also shows that networks that use leak batteries have longer link schedules than those that use leak-free batteries. Specifically, LSNBC with $|V| = 10, 20, 30, 40$, and 50 nodes produces superframes that are respectively 1.68, 5.81, 7.79, 10.06, and 10.48 times longer for networks with a leak battery as compared to when they use a leak-free battery. The reason is because with leakage, a battery needs a longer time to accumulate sufficient energy to transmit/receive a packet. The negative impact of using leak battery on schedule length is more apparent in LSBCC. More specifically, for $|V| = 10$ nodes, LSBCC produces 175 more slots (1.15 times longer) when the battery of nodes leak. Similarly for networks with 20, 30, 40, and 50 nodes, LSBCC generates superframes that are respectively 1.06, 1.03, 1.01, and 1.02 times longer for networks with a battery that leaks as compared to when they use a leak-free battery. This is reasonable because a battery that leaks means it will take a node a longer time to be charged to its maximum level and has a shorter discharging time period than a leak-free battery.

2) CHARGING *VERSUS* DISCHARGING CONSTRAINT

Figure 13 shows that in networks with batteries that leak, LSCC generates more slots than LSDC, which means charging constraint has larger effects on schedule length than the discharging constraint. More specifically, LSCC produces 0.67 (111 more slots), 0.65, 0.47, 0.41, and 0.77 times longer superframe than LSDC for networks with 10, 20, 30, 40, and 50 nodes, respectively, with standard deviation values ranging between 52 and 1190. This is because for LSCC the battery of each node can be charged if its capacity has reached the minimum level, while LSDC allows the battery to be charged at anytime. In contrast, for leak-free battery, LSDC produces schedules with more slots than LSCC. For example, LSDC generates 0.32 (32 more slots), 0.82, 0.92, 0.97, and 1.04 times longer superframe than LSCC for networks with 10, 20, 30, 40, and 50 nodes, respectively, with standard deviation values ranging between 31 and 144. The reason is because when the battery of nodes is leak-free, energy usage is only due to packet transmission/reception. That is, a battery does not reduce to its minimum level from energy loss due to leaking.

3) CHARGE/DISCHARGE CYCLES

In this simulation, we set $r_i = 5$ and $w_{i,j} = [1, 5]$; in addition, the leak rate of batteries is set to $\mu_i = \{1, 1.2, 1.4, 1.6, \cdots, 2.4\} \times 10^{-6}$. Figure 14 shows that the number of charge/discharge cycles produced by LSBCC is significantly less than LSNBC regardless of battery types. As an example, when nodes have a leak-free battery and there are $|V| = 10, 20, 30, 40, 50$ nodes, LSBCC generates 43.54% or 64 fewer cycles, 44.09%, 44.76%, 45.24%, 45.67% fewer cycles respectively than LSNBC, with standard deviation values ranging between 22 and 303. Similarly, when the battery of nodes leaks, LSBCC has 45.89%, 46.15%, 46.73%, 47.41%, and 47.11% fewer cycles than LSNBC for 10, 20, 30, 40, and 50 nodes, respectively, with standard deviation values ranging between 66 and 602. Due to the battery cycle constraint, each charge (discharge) occurs only when the battery has reached its minimum (maximum) energy level in LSBCC. On the other hand, when using LSNBC, a node's battery can be charged (discharged) at any slot when it is (is not) used.

Figure 14 also shows that the number of charge/discharge cycles when nodes have a leak-free battery is less than the case when battery leaks for both LSBCC and LSNBC.

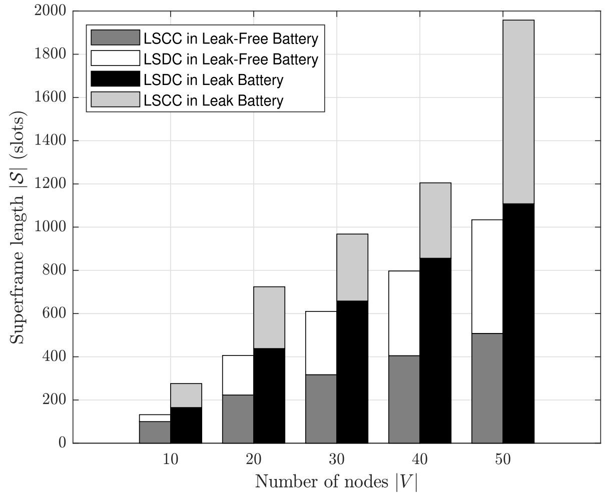


FIGURE 13. LSCC versus LSDC in networks where nodes have a leak-free and leak battery in terms of superframe length $|S|$.

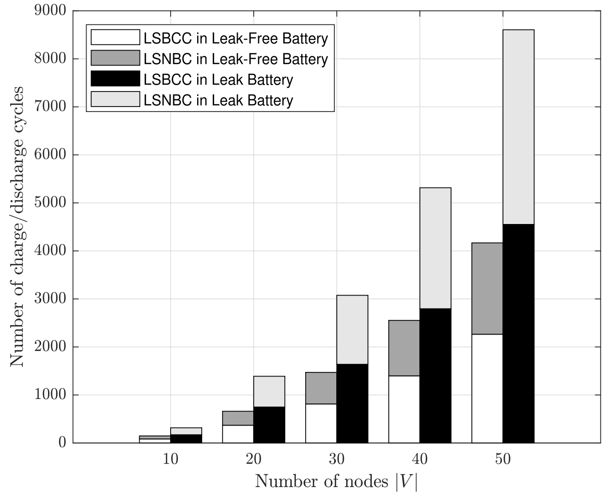


FIGURE 14. LSBCC versus LSNBC in networks where nodes have a leak-free and leak battery in terms of the number of charge/discharge cycles.

For example, for $|V| = 10$ nodes, LSBCC requires 88 or 51.46% fewer cycles when nodes are equipped with a leak-free battery than when they have a battery that leaks. The results are consistent for networks with 20, 30, 40, 50 nodes. More specifically, for the leak-free battery case, there are 50.67%, 50.43%, 49.98%, and 50.25% fewer cycles than when nodes have a battery that leaks. Similarly, LSNBC and nodes with a leak-free battery result in 53.48%, 52.48%, 52.2%, 51.97%, 51.57% fewer cycles than when they use a leak battery for $|V| = 10, 20, 30, 40, 50$, respectively. The reason is because a leak-free battery has less charging time interval than one that leaks.

V. CONCLUSION

This paper addresses a new link scheduling problem, called **LSME**, that considers the memory effects that degrade the lifetime of a node's battery. The problem is challenging as nodes have limited capacity, different charging times and leakage rates, and they use the HSU recharging model. As a solution, this paper proposes an algorithm called LSBCC. It also shows analytically the impact of enforcing a cycle constraint; a main finding is that if nodes' battery have a non-negative leakage rate, i.e., $\mu_i > 0$, the batteries at some end nodes may never be in the same discharging cycle. This finding is supported by extensive simulations, whereby the number of unscheduled links can be up to 98.81% of the total number of links in the network. When all links can be scheduled, enforcing cycle constraint increases superframe length by up to 1.71 times. However, it reduces the number of charge/discharge cycles by up to 47.41% as compared to cases where nodes do not have a battery cycle constraint. Further, LSBCC in networks with leak-free battery produces up to 0.54 (0.52) times shorter schedule (fewer charge/discharge cycle) than with leak battery. Our simulations also show that an increase in energy harvesting time linearly increases link schedules. Our work may result in unscheduled links because the batteries of their end nodes cannot be in the discharging state simultaneously. As future works, we plan to find a solution that guarantee all link schedules. Further, one may consider **LSME** that uses a dual alternate battery system. Such system is expected to reduce schedule length as well as address **LSME**'s link schedule feasibility.

REFERENCES

[1] J. A. Manrique, J. S. Rueda-Rueda, and J. M. T. Portocarrero, "Contrasting Internet of Things and wireless sensor network from a conceptual overview," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Chengdu, China, Dec. 2016, pp. 252–257.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. workshop Wireless sensor Netw. Appl. (WSNA)*, Atlanta, GA, USA, 2002, pp. 88–97.

[3] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal WSN deployment models for air pollution monitoring," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2723–2735, May 2017.

[4] K. S. Adu-Manu, C. Tapparello, W. Heinzelmamn, F. A. Katsriku, and J.-D. Abdulai, "Water quality monitoring using wireless sensor networks: Current trends and future research directions," *ACM Trans. Sensor Netw.*, vol. 13, no. 1, pp. 4:1–4:41, Feb. 2017.

[5] V. J. Hodge, S. O'Keefe, M. Weeks, and A. Moulds, "Wireless sensor networks for condition monitoring in the railway industry: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1088–1106, Jun. 2015.

[6] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[7] H. Kim, Y. Tadesse, and S. Priya, *Piezoelectric Energy Harvesting, Energy Harvesting Technology*, S. Priya and D. J. Inman, Eds. New York, NY, USA: Springer, 2009.

[8] *MICA2 Datasheet—Crossbow Technology*. Accessed: Feb. 12, 2018. [Online]. Available: https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

[9] R. Huggins, "Mechanism of the memory effect in 'Nickel' electrodes," *Solid State Ionics*, vol. 177, nos. 26–32, pp. 2643–2646, Oct. 2006.

[10] M. I. Hlal, V. K. Ramachandaramurthy, A. Sarhan, A. Pouryekta, and U. Subramaniam, "Optimum battery depth of discharge for off-grid solar PV/battery system," *J. Energy Storage*, vol. 26, Dec. 2019, Art. no. 100999.

[11] J. Yang and J. Wu, "Optimal transmission for energy harvesting nodes under battery size and usage constraints," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 819–823.

[12] G. M. Siddesh, G. C. Deka, K. G. Srinivasa, and L. M. Patnaik, *Cyber Physical Systems—A Computational Perspective*. Boca Raton, FL, USA: Chapman & Hall, 2015.

[13] 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, Aug. 2008.

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

[15] Tony, S. Soh, M. Lazarescu, and K.-W. Chin, "Link scheduling in rechargeable wireless sensor networks with harvesting time and battery capacity constraints," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, Chicago, IL, USA, Oct. 2018, pp. 235–242.

[16] 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. 104721–104736, 2019.

[17] 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. 144027–144038, 2020.

[18] T. Tony, S. Soh, K.-W. Chin, and M. Lazarescu, "Link scheduling in rechargeable wireless sensor networks with battery memory effects," in *Proc. Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Melbourne, VIC, Australia, Nov. 2020, pp. 1–8.

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

[20] 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.

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

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

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

[24] G. E. Andrews, "The geometric series in calculus," *Amer. Math. Monthly*, vol. 105, no. 1, pp. 36–40, Jan. 1998.

[25] T. Dillig. (2012). *Mistral SMT Solver*. Accessed: Jul. 30, 2020. [Online]. Available: https://www.cs.utexas.edu/ tdillig/mistral/index.html

[26] I. Dillig, T. Dillig, and A. Aiken, "Cuts from proofs: A complete and practical technique for solving linear inequalities over integers," in *Proc. Int. Conf. Comput. Aided Verification*. Berlin, Germany: Springer, 2009, pp. 233–247.

[27] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, p. 32, Sep. 2007.

[28] A. J. Bondy and U. S. R. Murty, *Graph Theory with Applications*. London, U.K.: Macmillan, 1976.

TONY TONY (Member, IEEE) received the bachelor's degree in computer science from Tarumanagara University, Indonesia, in 2005, and the M.C.S. degree from the University of Indonesia, in 2010. He is currently pursuing the Ph.D. degree with Curtin University, Perth, Australia. His research interests include link scheduling and wireless sensor networks.

SIETENG SOH (Member, IEEE) received the B.S. degree in electrical engineering from the University of Wisconsin–Madison, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Louisiana State University, Baton Rouge, in 1989 and 1993, respectively. From 1993 to 2000, he was with Tarumanagara University, Indonesia. He is currently a Senior Lecturer with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, Australia. He has published over 100 international conference and journal papers. His current research interests include algorithm design, network optimization, and network reliability.

KWAN-WU CHIN received the B.Sc. (Hons.) and Ph.D. degrees in commendation from Curtin University, Australia, in 1997 and 2000, respectively. He was a Senior Research Engineer with Motorola, from 2000 to 2003. In 2004, he joined the University of Wollongong, as a Senior Lecturer, where he is currently an Associate Professor. His research areas include medium access control protocols for wireless networks, and resource allocation algorithms/policies for communications networks. He currently holds four U.S. patents and has published more than 150 conference and journal papers.

MIHAI LAZARESCU (Member, IEEE) received the B.S. (Hons.) and Ph.D. degrees in computer science from Curtin University, Perth, Australia, in 1996 and 2000, respectively. He has been a Senior Member of the IMPCA Research Institute for ten years. He is currently the Computing Discipline Lead and an Associate Professor with Curtin University. He has published over 80 papers in refereed international journals and conference proceedings in the areas of artificial intelligence, machine vision, data mining, and network reliability.