ReNEW: A Practical Module For Reliable Routing in Networks of Energy-harvesting Wireless Sensors

Vijay S. Rao, *Member, IEEE*, R. Venkatesha Prasad, *Senior Member, IEEE*, Chayan Sarkar, *Member, IEEE*, Ignas Niemegeers.

***Abstract*—It is a huge challenge to run IoT devices/sensors powered solely through ambient harvested energy. Since the harvested energy is less and is stochastic in nature, it is extremely challenging to achieve low latency and high reliability. To this end, we propose a distributed, energy-management module called ReNEW, using Constructive Interference (CI) to achieve our target of increased reliability, especially in the low harvesting regimes. We choose CI-based protocols to leverage low latency guarantees. Specifically, we propose a Markov-Decision model to maximize the energy utility in the infinite horizon by allocating energy optimally using a threshold-optimal policy. Since an energy scheduler is insufficient we propose distributed techniques to conserve energy on redundant nodes in the network, and dynamically activate them based on feedback. We implement ReNEW on Indriya and FlockLab testbeds for real-world scenarios in a network of 20 source nodes out of the 30 nodes. ReNEW collects data periodically with 2.5 times higher packet reception compared to LWB when the harvested energy is as low as 50μJ/s for 100B packets every 30s with a saving of 25% higher residual energy. In a nutshell, by integrating ReNEW with CI based protocols, we enable guaranteed latency and increased reliability in battery-less devices/networks.**

***Index Terms*—Energy-harvesting, Wireless Sensor Networks, IEEE 802.15.4, Constructive Interference, Dynamic Activation, Power Allocation**

I. INTRODUCTION

Many Internet of Things (IoT) applications require low latency and high reliability[1] to enable closed-loop control [1]. Low end-to-end latency, high reliability and long lifetime of the network are the parameters that determine the usability and success of the IoT deployment.

Batteries limit the lifetime of the devices and in turn the utility of the network and the applications. Powering all the IoT devices through batteries is not scalable as frequent battery replacement is either labor-intensive or impractical due to physical or deployment conditions [2]. Thus many IoT infrastructures adopt Energy-Harvesting Wireless Sensor Networks (EH-WSNs). As batteries are unsustainable, we only consider nodes powered with energy storage buffers such as supercapacitors.

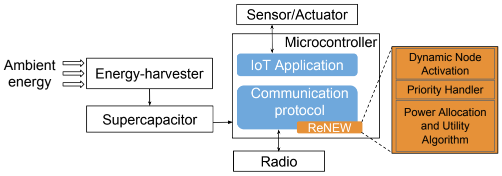


Fig. 1: The ReNEW energy-management module.

A key problem is to guarantee high reliability and low latency in EH-WSNs such that these parameters (reliability and latency) satisfy the application requirements. This is particularly challenging when data dissemination and collection are periodic in EH-WSNs. Ambient-energy sources do not provide constant power, and the harvested energy from different sources varies drastically over location and time [3]. Given the stochastic nature of energy arrivals, existing networking protocols for EH-WSNs target only reliable packet delivery [4], [5], [6] by adapting to the variations in energy for longer lifetimes rather than also ensuring low latency. Furthermore, they may suffer from Braess paradox [7], wherein the high energy nodes attract more traffic leading to their death. On the other hand, routing protocols have been defined since two decades for battery-powered WSNs that target achieving both guaranteed reliability and latency. Particularly, Constructive Interference (CI) based protocols [8], [9] have been shown to collect and disseminate data in a highly energy-efficient and reliable manner with low latency for periodic traffic. However, they fail in EH-WSNs due to the dynamic energy variations. The most plausible conclusion from the current literature is that the EH-WSNs cannot support low latency operations, at least to a reasonably satisfactory extent. Thus, the ambition is to avoid overheads, achieve low latency and high reliability under challenging conditions, i.e., low energy-harvesting conditions.

Approach. CI mechanism eliminates the need for contention to access the wireless medium. CI occurs when two or more nodes transmit the same data concurrently, which makes the signals superpose. Hence, receivers can decode the packet successfully with high probability due to the increased signal power at the receivers. Recently, Ferrari et al. [12] made a major contribution through their flooding technique called Glossy that exploits CI. Glossy achieves latency close to the lower theoretical limit and also implicitly synchronizes the nodes with sub-microsecond accuracy with high reliability. Low-power Wireless Bus (LWB) builds on Glossy to achieve a low-power, low-latency, and reliable data collection and

---

This article was presented in part at the 2019 IEEE Global Telecommunications (GLOBECOM), Waikoloa, Hawaii, USA.

V. Rao and R. Venkatesha Prasad are with the Embedded and Networked Systems group of the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. E-mail: {V.Rao, R.R.VenkateshaPrasad}@tudelft.nl

C. Sarkar is with Robotics & Autonomous Systems, TCS Research and Innovation Labs, India.

I. Niemegeers is with Eindhoven University of Technology, The Netherlands.

[1] We define reliability as the Packet Reception Ratio (PRR), which is the percentage of packets that are successfully received at the destination/sink.

Name	Storage	Working Principle	Basic Idea	Node Wakeup	Reliability	Latency Guarantees
CTP [10]	Battery	Tree-based	Nodes select parents with lower routing cost and ETX	Asynchronous	High	Yes With increased duty cycle
Dozer [11]	Battery	Time-slotted	Nodes select parents with lower hop-count and load	Scheduled	High	No, collisions cause delays
LWB [8]	Battery	CI based	Every packet is flooded	Scheduled	High	High
ORiNoCo [6]	Super-capacitor	Opportunistic with receiver initiated MAC	Nodes send packets to beacons with low routing cost. High energy nodes wakeup more often	Asynchronous	High	No
EHOR [4]	Super-capacitor	Opportunistic	Routing metric is a function of residual energy and hop-count	Asynchronous	High	No
SP-BCP [5]	Rechargeable battery	Back-pressure	Backpressure calculation is made harvesting energy aware	Asynchronous	Medium to high	No

TABLE I: Summary of available routing protocols for WSNs and EH-WSNs

dissemination protocol by scheduling timeslots to each sensor node such that the data from the sensor node is flooded using Glossy in its timeslot. LWB divides its communications into rounds, which are periodically scheduled. Each round begins with the dissemination of the schedule, using Glossy, for that round by the sink. This is followed by each sensor node taking turns to become the 'initiator' in Glossy to flood its packet in its allocated slot.

As CI based protocols offer an energy-efficient platform that guarantees low latency (almost close to the theoretical limit), we focus the work only on providing reliability for these protocols. To this end, we propose an energy-management module called ReNEW (*Re*liable routing in *N*etworks of *E*nergy-harvesting *W*ireless sensors) to enable high reliability in EH-WSNs. To prove our point, we use Low-power Wireless Bus (LWB) [8] as the *de facto* routing protocol and develop ReNEW around it. LWB offers guaranteed latency and high energy efficiency without any topological information. While high reliability is also guaranteed by LWB in battery-powered WSNs or CI based protocols in general, it remains a non-trivial challenge in EH-WSNs as nodes do not have sufficient energy as required by LWB. In particular, in low energy harvesting campaigns, nodes need to be intelligent to use the available energy wisely.

Challenges. The main challenges that ReNEW must overcome are as follows:

1) Nodes do not have the energy to participate in all communication slots. This leads to low reliability or packet reception ratio (PRR).
2) An energy-aware scheduler or duty-cycling mechanism on each node is insufficient, the network as a whole may be wasting resources. The network may suffer when the nodes are harvesting less as they become highly conservative in their participation, which also implies that the benefits of CI to overcome the unreliable wireless channel are lost.

We address these challenges in this paper. The crux of ReNEW is to achieve better reliability through efficient energy-management strategies. Fig. 1 shows the ReNEW module and its components. ReNEW maximizes the energy utilization by:

(a) allocating an optimal amount of energy to spend in every data transmission slot,
(b) spending energy on the most important slots,
(c) saving energy by reducing transmission power as and when possible, and
(d) utilizing the redundant nodes deployed in the network.

In ReNEW, every node wakes up to receive the schedule by the sink at the start of the communication round. Based on the available energy and the incoming energy, a node decides how many slots it can participate in and corresponding energy for the round is allocated. Note that this is a distributed algorithm, i.e., each node decides on its own action without any coordination with other nodes. The energy allocation problem is modeled as a Markov Decision Problem in the infinite horizon to obtain a low-complexity optimal policy. Then, the allocated energy is used in as many slots as possible starting from the highest priority ones. Priority of a slot is determined based on the feedback from the sink; the highest priority is given to a slot in which the packets were not received at the sink with the node not participating in the slot where it should have and the lowest priority slot is the one where packets were received at the sink with the node not participating in the slot. Furthermore, instead of transmitting all the packets at the highest transmission power, the node will reduce its transmission power if packets are being delivered successfully. The advantages are that energy is being better used and improve the performance of the underlying CI phenomenon [13]. Specifically, our contributions are as follows:

- To the best of our knowledge, this is the first work (apart from our conference publication [14]) that attempts to provide guarantees on latency and improves reliability considerably for data collection and dissemination in EH-WSNs. This practically important aspect is novel and has not yet received its due attention. To this end, we propose a distributed, energy-management module called ReNEW.
- We formalize the energy allocation problem as a Markovian decision problem and we propose a threshold optimal policy.

- We propose a set of protocol optimizations in ReNEW to make better use of the available redundant nodes and increase the performance of CI in the network.
- We implement and evaluate the performance of ReNEW on Indriya and FlockLab, two real-world testbeds with CC2420 radios [15], [16] for real-world scenarios considering different number of nodes and data collection intervals.

We show that in one of the worst-case scenarios – where harvested energy rate is as low as 50μJ/s with 20 nodes in the network with the transmission of 100 B every 30 s – we even get an improvement of 2.5 times higher packet reception ratio, with 6 mJ higher remaining energy on the average compared to the LWB based greedy algorithm.

Organization. We present the related work in Sec. II. We provide an overview and benefits of LWB in Sec. III. Sec. IV provides an overview of ReNEW. Then, we solve the energy allocation problem in Sec. V. Further, we describe the protocol optimization in Sec. VI. Sec. VII evaluates and discusses the performance of ReNEW and conclude the article in Sec. VIII.

II. RELATED WORK

The work on routing in EH-WSNs has attracted less attention compared to their battery-powered counterparts. Table I summarizes the most significant networking protocols in WSNs and EH-WSNs. CTP, Dozer, and LWB have been proposed and optimized for battery-powered WSNs wherein the nodes can wake up regularly until the battery dies. Collection tree protocol (CTP) [10] is a well-known protocol for data collection in WSNs. In this protocol, a tree is built, with the sink as the root, across all nodes in the network. Nodes select parents with low link and routing costs. Dozer [11] is another energy-efficient protocol for periodic data collection in WSNs. Dozer reduces idle-listening and overhearing and incorporates a cross-layer solution that spans across MAC layer, topology control, and routing protocols. It builds a data gathering tree on the underlying topology and uses timeslots for data transmission based on local synchronization. Low-power Wireless Bus [8] is based on Glossy, a CI based flooding primitive, for periodic data collection. LWB is a topology-free and highly energy-efficient data collection and dissemination mechanism. LWB is shown to be highly energy-efficient than both CTP and Dozer.

ORiNoCo [6] is an opportunistic routing protocol for EH-WSNs that builds upon receiver-initiated MAC. Nodes wake up when they have sufficient energy and send beacons when ready to receive. Higher energy nodes wake up more often than their lower energy counterparts. The neighboring nodes that are awake and have a packet to send are required to respond to the beacons. Nodes choose to respond to the beacons that offer low routing costs to the sink. EHOR [4] is another opportunistic routing protocol for EH-WSNs that creates a gradient towards the sink. The gradient is created using a routing metric that is a function of the residual energy and the hop-count to the sink. SP-BCP [5] is a backpressure routing algorithm that is adapted for EH-WSNs. While these algorithms are energy-harvesting aware and offer high reliability in data collection, they are not suited for periodic data collection applications and do not provide any guarantees on latency.

Of these limited works, most of them such as ORiNoCo [6] (opportunistic receiver-initiated no-overhead collection protocol) and SP-BCP [5] (solar-powered backpressure collection protocol) target reliably delivering packets to the sink through higher energy nodes. The reasons for not targeting low latency in EH-WSNs are: (a) energy variations make it difficult to get the nodes globally synchronized as traditional synchronization protocols are power-hungry; (b) schemes such as Low Power Listening still have a considerable amount of overheads before successfully transmitting data, and (c) packet losses on the wireless channel consume a significant amount of energy for retransmissions. Furthermore, ORiNoCo and EHOR suffer from Braess's paradox [7], wherein the gradient created towards higher energy nodes turns detrimentally. These higher energy nodes may deplete energy faster leading to lost data packets.

A common strategy employed to make WSN protocol energy-harvesting aware is by using power-management techniques such as adaptive duty-cycling, scheduling tasks, and transmission policies. However, directly using them on LWB will not render the desired features. Adaptive duty-cycling techniques [17] determine how long a node should be awake based on residual energy and energy harvesting rates. While these algorithms can be tweaked to determine how much energy to spend, they do not schedule the operation of tasks. We show this in Sec. VII as we compare ReNEW to the adaptive cycling mechanism proposed in [18]. Task scheduling [19] algorithms, on the other hand, maximize the number of tasks executed within some specified deadlines by considering the energy remaining in the storage element. However, these algorithms are myopic in their approach.

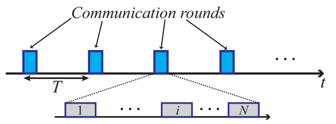
Markov models representing energy availability have been proposed to determine optimal transmission policies [20], [21], [22]. Each packet to be transmitted is considered to have a certain value, and the node gets a reward proportional to this value if the packet is transmitted. On similar lines, transmission power policies have also been constructed [20]. Rewards accrued are proportional to the energy state. These models target maximizing the average reward over an infinite horizon, which implies that the node will optimize its energy usage and packet transmissions. Other works include resource and power allocation policies [23], [24] for EH networks. These works also cannot be used since they either schedule packet transmission in a future time when the energy is higher or do not consider transmission power to improve the performance of CI.

III. OVERVIEW OF LOW-POWER WIRELESS BUS (LWB)

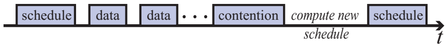
We provide an overview of the working of the Low-power Wireless Bus (LWB) protocol before we discuss ReNEW. We refer the interested reader to [8] for a detailed explanation. Since LWB is built on top of Glossy [12], which is one of the recent path breaking works in WSNs, we first provide a brief overview of Glossy.

A. Glossy

When two nodes transmit the same packet simultaneously on the same frequency band to a receiver within their trans-



(a) Communication rounds in LWB.



(b) Communication slots within a round.

Fig. 2: Time-triggered operation of LWB.

mission ranges, the transmitted signals superpose leading to constructive interference (CI) at the receiver. CI can aid in increasing the decodability of the packet due to increased energy per symbol at the receiver. The tolerable temporal displacement between concurrent transmissions for IEEE 802.15.4 in 2.4 GHz is $0.5\,\mu$s.

Glossy uses CI for flooding and implicitly provides time-synchronization. In Glossy, an initiator node has packets that must be disseminated in the network. The initiator first sends a packet that is received by the first-hop nodes. The nodes turn on their radios, listen for packets on the wireless medium, and relay the received packets immediately after receiving them. Since all the potential receivers receive a packet at the same time, they also start to relay the packet at the same time. This again triggers other nodes in the next hop to receive and relay the packet. In this way, Glossy benefits from concurrent transmissions and quickly propagates a packet from a source node (initiator) to all other nodes (receivers) in the network as a ripple. Every packet is transmitted η times (the default value is five), in order to ensure high reliability. Note that all events are initiated by radio events. Since the medium contention is eliminated, Glossy achieves very low-latency flooding, and nodes are synchronized in the process.

Glossy floods are periodic. Therefore, each node calculates the next time it has to wake up. Furthermore, there is only one data packet that is disseminated in the network at any given instant. Since nodes just transmit the packets that they received, the nodes do not contend to access the channel. Therefore, Glossy avoids idle listening, collisions, and overhearing. The overheads are demonstrated to be minimal as well since the protocol information is combined in the data packet, which reaches all the nodes at the end of the round. This makes Glossy a highly energy-efficient, low-latency, and highly reliable flooding mechanism.

B. Low-power Wireless Bus (LWB)

LWB uses the Glossy fast-flooding primitive to deliver data. Since the packet is flooded throughout the network, it eventually reaches the destination(s). In order to avoid packet collisions from different source nodes within the same flood period, LWB uses a centralized scheduler to enable TDMA.

LWB divides its communications into rounds. Communication rounds are periodically scheduled, which is shown in Figure 2(a). The communication slots within a communication round in LWB is shown in Figure 2(b). Each round begins with the dissemination of the schedule, using Glossy, for that round by the sink. This is followed by each sensor node taking turns to become the ‘initiator’ in Glossy to flood its packet in its allocated slot. The scheduler usually runs in the sink node, assigns a unique slot to every data source within a communication round, and only the slot owner initiates flooding in that slot. Nodes requiring a slot will use the contention access period to send in their requests.

Since every packet is flooded, LWB does not require any topological information. The analogy here is similar to a bus, wherein one node (initiator) puts the data on the bus that can be read by all other nodes. All nodes participate in all the floods to exploit CI.

C. Existing Optimizations for LWB

Several optimization schemes are proposed for LWB to reduce energy consumption. The important ones are listed below.

Long-run conditions: During the bootstrap phase, the sink can learn about the source nodes and data periodicity. After a certain duration, the traffic pattern can stabilize. In such situations, LWB minimizes the overheads by increasing the round-trip time without violating the maximum latency that can be tolerated.

Forwarder Selection: While in classic LWB, all the nodes in the network need to forward data towards the destination, an optimization was proposed by Carlson et al. [25], wherein each node decides whether it falls in the shortest path between a source and destination pair. To determine the shortest path, a packet is flooded from the source to the destination and also in the reverse direction. Each packet contains hopcount information. By comparing the hopcounts received by a node and the shortest hopcount between the source and the destination, the node can easily determine if it lies in the shortest path.

Nodes not in the shortest path need not participate in floods. This reduces energy consumption and unnecessary flooding of packets in the entire network. A small overhead is incurred as the set is determined by flooding messages between the source and destination nodes.

Apart from the above, minor improvements could be done, such as nodes piggybacking their requests for additional slots in their data packets to reduce contention and the contention period during long-run conditions. These improvements, however, are *highly insufficient* on nodes powered by harvesting energy.

Advantages of LWB

1) LWB can achieve latencies close to theoretical limits due to concurrent transmission based flooding. The average latency for several hops from our experiments is shown in Table II.

2) Due to concurrent transmissions, medium access contentions are eliminated. Furthermore, due to scheduling each slot, and each node being synchronized to an accuracy of $0.5\,\mu$s, idle-listening is also eliminated. This results in the high energy efficiency of the protocol. It has been shown that LWB outperforms *de facto* protocols such as CTP [10] and Dozer [11] for periodic data collection scenarios.

3) LWB has been shown to offer high reliability on unreliable wireless channels due to CI.

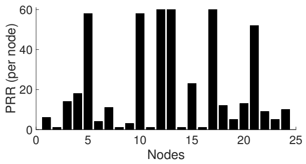


Fig. 3: Packet Reception Ratio (PRR) per node after 60 rounds.

4) Due to flooding, topological information is not required anymore making LWB work even when the nodes are mobile.

D. Challenges in using LWB for EH-WSNs

While LWB seems ideal, it does not work out of the box for EH-WSNs. We experimented with 25 Tmote Sky nodes wherein each node harvested energy in 'packets' (of 50 μJ) according to the uniform distribution. The storage buffer could store up to a maximum of 180 mJ. We measure the reliability through packet reception ratio (PRR) at the sink. Fig. 3 shows the PRR for a periodic collection of every 60 s from every node after 60 rounds. It is evident that the performance is not acceptable barring a few nodes. The reasons for such a performance are due to

- Not enough energy on the nodes to participate in all the slots, sometimes even for a node's own data transmission slot.
- Each packet is sent five (η) times in order to overcome the unreliable wireless channel, which leads to draining the energy faster.
- Unequal energy-harvesting opportunities leading to some nodes having good performance and many others not.

Therefore, we propose to ReNEW to manage energy wisely in energy-harvesting WSNs. In the next section, we present an overview of the ReNEW module, its design, and operation.

IV. AN OVERVIEW OF THE RENEW MODULE

We propose to ReNEW to manage the energy wisely in EH-WSNs in a distributed manner. The crux of ReNEW is to achieve better reliability through efficient energy-management strategies. Fig. 1 shows the ReNEW module and its components. ReNEW maximizes the energy utilization by (a) allocating an optimal amount of energy to spend in every data transmission slot, (b) spending energy on the most important slots, (c) saving energy by reducing transmission power as and when possible, and (d) utilizing the redundant nodes deployed in the network. In EH-WSNs, it is a practical requirement to deploy redundant nodes [26]. ReNEW has three major components, namely, dynamic node activation, priority handler, and energy allocation and utility. Note that ReNEW is fine-tuned for LWB but can be easily extended to any CI based protocol, and with a little work to any slot-based communication protocol for EH-WSNs. Furthermore, LWB can schedule up to 120-odd source nodes in one round. If there are more source nodes, then more rounds can be scheduled by selecting a different set of sources in each round. However, this can penalize the nodes on their energy, especially as all nodes need to participate in every round. A more effective alternative is to use a protocol such as Fleet [24], which can be scaled to 1000s of nodes. In Fleet, the network is divided into clusters, wherein each cluster-head runs LWB or CI based protocol within its zone. ReNEW can easily be adopted for Fleet without major changes.

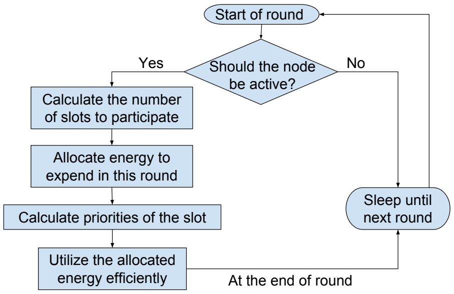


Fig. 4: Working of the ReNEW energy-management module.

Fig. 4 shows the flowchart of ReNEW. At the beginning of a communication round, the node wakes up and determines if it needs to be active in this slot. This is pertinent to the redundant nodes to ensure connectivity when the energy-harvesting possibility is low. The source nodes, however, will choose to be active as long as they have sufficient energy in their buffer. Nodes choosing to be inactive go to sleep until the next round. This is described in Sec. VI.

The next step for the active nodes is to note down the slots to participate from the LWB schedule. After that, the energy allocation module has two questions to address: *how much energy to expend in the current round?* and *how to utilize this energy efficiently?* ReNEW looks at maximizing the node's utility over an infinite horizon. The former question is addressed in Sec. V wherein an optimal allocation policy is proposed. For the latter question, we tweak LWB to provide feedback from the sink and use it to prioritize the slots using the priority handler. Then, the allocated energy is used in as many slots as possible starting from the highest priority ones. Protocol optimization is proposed here: instead of transmitting all the packets at the highest transmission power, the node will reduce its transmission power if packets are being delivered successfully. The advantages are that energy is being better used and improve the performance of the underlying CI phenomenon (see Sec. VI). With all these components and protocol optimization, the nodes utilize the available energy with higher efficiency as will be shown in Sec. VII. In the next section, we formalize the power allocation problem and provide an optimal policy for the same.

V. ENERGY ALLOCATION

In this work, we consider LWB with forwarder selection since it is already an improved version of LWB. Henceforth, when we refer to a node that should participate in a slot implies that the slot is either one of the forwarder selected or its own slot. While the slot schedules are distributed from the sink, each node will have to manage its energy expenditure on its own. Every node must adopt an energy-aware policy to balance the available energy for expenditure in the future and in the current slot. In this section, we address the question:

How much energy should be expended in the current time period? Intuitively, if a node aggressively participates in all its forwarder selected slots, the energy gets depleted soon. On the other hand, if the node is too conservative, then the PRR is low because of its non-participation within the network. To this end, we propose to use the Markov Decision Process (MDP) framework. Though there have been several works that propose to use MDP for determining the optimal transmission policies per packet [21], [20], we differ from these works in the following aspects: (i) we cannot 'queue' slots for the future as in some of those models and (ii) we do not decide to transmit in a particular slot but rather allocate energy for the whole communication round.

A. System Model

We consider an EH-WSN network consisting of N nodes with omnidirectional antennas. Every node u in the network has a unique identifier, denoted as $id(u)$. As we target a distributed algorithm, we focus on a single sensor node. We consider that the harvested energy between the communication rounds k and $k+1$ follows an *i.i.d.* process represented by $Y(k)$ (e.g., [27]). Each node has a supercapacitor as a storage buffer. The sink node is connected to a power grid.

We assume that communication slots for the node are modeled with an arrival process, $X(k)$, and also follows *i.i.d.* Let the number of slots to be allocated in the k^{th} round be $x(k)$. A decision must be made as to how many of these slots will be allocated energy. The remaining slots will be discarded. We model the energy buffer by quantizing it into states $\varepsilon = \{E_0, E_1, \dots E_{max}\}$. Each state holds energy enough for one slot with maximum transmission power (including transmitting for η times). The energy for round $k+1$ can be computed as,

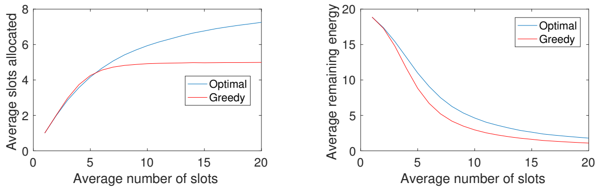
$$E(k+1) = \min\{E(k) - A(k) + Y(k), E_{max}\}, \tag{1}$$

$A(k)$ is the energy allocated in k. The slot arrival process follows $x(k+1) = X(k)$. If the sink follows a static schedule, $X(k)$ can be defined to be uniform distribution with the parameters configured tightly to keep the number of slots constant. This does not affect the model and is compliant with the real-world scenario. We consider a concave, monotonically non-decreasing function, g with $g(A(k))$ indicating the number of slots allocated if $A(k)$ amount of energy is used.

B. The Optimization Problem and an Optimal Policy

Given a state $E(k) \in S$, value $v(k) \in \mathbb{R}^+$, a policy π implemented by the node is defined by the probability $\pi(\varepsilon, v)$ of selecting $x(k)$ slots in the communication round k. The optimization problem can be formulated as a Markovian decision problem wherein we need determine the optimal policy π_* such that $\pi_*(s) = \arg\max_\pi V^\pi(s_0)$, where s_0 is the initial state and V^π is the value of the policy.

Optimal policy. The necessary condition for an optimal policy is: For $\{A(k)\}$ to be asymptotically stationary, a policy that makes $\{x(k)\}$ asymptotically stationary with a stationary distribution π, it is necessary that $\mathbb{E}[X] < \mathbb{E}_\pi[g(A)] \leq g(\mathbb{E}[Y])$ [28].



(a) (b)

Fig. 5: (a) Average number of slots allocated for a varying average number of slots arrivals. (b) Average remaining energy for a varying average number of slot arrivals.

We present a policy that satisfies this condition. Let

$$A(k) = min(E(k), \mathbb{E}[Y] - \epsilon), \tag{2}$$

where ϵ is a small positive constant with $\mathbb{E}[X] < g(\mathbb{E}[Y] - \epsilon)$. This is indeed an optimal policy as a stationary (or a threshold vector) does exist as this satisfies the necessary condition. Asymptotically, as g is concave, $g(A(k)) \rightarrow g(\mathbb{E}[Y] - \epsilon)$. Thus, $\{g(A(k))\}$ is asymptotically stationary and ergodic. Thus $\mathbb{E}[X] < g(\mathbb{E}[Y] - \epsilon)$ is a sufficient condition for $\{x(k)\}$ to be asymptotically stationary and ergodic whenever $\{X(k)\}$ is stationary and ergodic [28].

Intuition. The policy indicates that all slots be allocated if the average harvested energy is higher than the required energy. In case, the required energy becomes more, then only the average amount of energy harvested will be spent. Therefore, the number of slots that are allocated will be $g(\mathbb{E}[Y] - \epsilon)$. This ϵ may represent the minimum energy required to at least send data in its own slot. Therefore, all the nodes will try to be active as much as possible in the infinite horizon.

C. Numerical Results

We evaluate the optimal policy by comparing it with a greedy policy as given in Eqn. 3. The intuition is the greedy policy attempts to allocate energy for participating in the maximum number of slots possible with the available energy. We evaluate the policies through numerical simulations.

$$A(k) = min(x_e(k), E(k)), \tag{3}$$

where $x_e(k)$ is energy required for participating $x(k)$ slots.

We consider g to be linear ($g(x) = x$), which is monotonically non-decreasing. Furthermore, $X(k)$ (slot arrival process) and $Y(k)$ (energy arrival process) are *i.i.d.* with exponential and uniform processes, respectively. We choose the uniform distribution as it models an indoor ambient light harvesting source [29]. Without the loss of generality, we set one unit of energy to participate in a slot. Lastly, we set $\mathbb{E}[Y] = 5$ with the maximum energy storage size of 20 units, and $\mathbb{E}[X]$ is varied from 1 to 20.

Fig. 5 shows the results for two metrics: average number of slots allocated and average remaining energy in the node. Fig. 5(a) shows that the node can participate in more slots by using the optimal energy allocation than the greedy policy. The reasoning is simple and obvious: while the greedy policy tries to participate in as many slots as possible at the cost of energy exhaustion, the optimal policy is energy aware and adapts its expenditure according to the energy being harvested. This increases the utility of the node in the infinite horizon. Fig. 5(b) shows that both the policies spend almost the same amount of energy to participate in a slot, while optimal policy makes better use of the energy.

VI. PROTOCOL OPTIMIZATION

While we saw that the optimal policy outperformed the greedy policy, we notice that only 7 out of 20 slots were assigned to transmit data. This is due to the amount of energy harvested being quite low compared to the consumption rate. To handle such situations, we propose several solutions.

Dynamic Node Activation. Since the available energy on the nodes is quite low, a commonly adopted solution is to deploy redundant nodes [26]. A redundant node does not generate any sensor data of its own, i.e., it is not a data source. Therefore, it does not request for any data slots for itself. This is particularly helpful when there are no secondary power sources, such as batteries, as these nodes can help flood packets in the network.

The purpose of redundant nodes is not served if all the nodes, including the redundant nodes are always on. These "helper" nodes must be dynamically switched on when required. Though the authors of [26] propose policies to activate nodes, it is assumed that the redundant nodes can check the neighborhood status. Such an assumption does not hold in our scenario. Therefore, we design a simple distributed policy. A non-source node is activated according to the policy given in Eqn. 4 for a communication round k on a node i. A source node is always activated if it has a minimum amount of energy, E_{min} to at least participate in its own slot.

$$\mathcal{A}_i(k) = \begin{cases} \text{no activation} & \text{if } E(k) < E_{min} \\ \text{activate with prob. } p & \text{if } E_{min} < E(k) \leq E_{th} \\ \text{activate with prob. 1} & \text{if } E(k) > E_{th} \end{cases} \tag{4}$$

Priority Handler. Since the nodes may not always have sufficient energy to participate even in all its forwarder selected slots, it is important to quantify the importance of slots. By defining weights, the nodes can then choose the best slots to participate. The priority handler ensures that the energy is spread across the slots and not spent on the first few slots (as in the greedy approach).

A difficulty though is that individual feedback cannot be given to the nodes. We tweak the LWB protocol to make the sink include the information on which slots data was successfully received in the previous communication round. This information, or *ACK*, is piggybacked with the following communication round's schedule. With this ACK information, the node has four cases to deal with:

- The best case is if a node participated in a slot and the packet was received. The priority must be slightly increased in this case so that the node is more likely to participate in the slot again.
- Another case is when the node participated in forwarding data in a slot but was not received at the sink due to failure of CI or an energy outage at another intermediate node. Here, the node cannot do much but try to participate again.
- If a node sees that ACK is received in a slot it did not participate, then the node decrements the priority since its participation is not required for successful data delivery.
- The worst case is when a slot goes unserved i.e., the node did not participate and the data did not reach the sink as well. In this case, the node assumes responsibility by increasing its priority to a higher value.

One method to calculate the weight is to take (1 - PRR) per slot. We increment or decrement priority by 10% of its value.

Energy utilization. The optimal policy only allocates the energy but does not specify how to use it. With priorities defined to the slots, the problem becomes that of allocating the energy to as many high-priority slots as possible. This can be proven to be the classical 0/1 knapsack problem [30]. As the 'weights' of each slot are the same, this problem can be solved in polynomial time ($O(NlogN)$). The slot assignment algorithm is shown in Alg. 1. In order to save energy, we lower the transmission power when the transmissions happen successfully. The advantages are two-fold: (a) employing different transmission powers across nodes improves the performance of CI [13]. (b) If enough power is saved to serve more slots, then the next higher priority slots are chosen to participate in.

To this end, we modify DIPA [13]. The philosophy of DIPA is to exploit the varying transmit powers of the nodes for achieving a better performance of CI. The intuition of DIPA is to increase transmit power if packets are not being received, and slowly decrease if the packet reception is stable. As ACKs are not possible in CI based protocols, DIPA incorporates a feedback byte in the packet. If a packet is received correctly, the next packet transmitted will contain ACK in the feedback byte, NACK otherwise. In order to retrieve the packet data and the feedback byte in DIPA, a software-based CRC computation is required.

While this is possible, the source code needs to be optimized to ensure timing constraints are met. We simplify DIPA for the ease of implementation in this work: there is no feedback byte and we enable auto CRC, i.e., hardware-based CRC checks. We leverage the fact that CI based transmissions are also received by the sender when the next hop retransmits the packets as feedback.

Consider the following scenario: An initiator transmits a packet, which is received by the next hop nodes. When these nodes retransmit the packet immediately, the initiator also receives the packet. If this packet is received correctly, i.e., CRC is valid, then the transmit power is decreased. Otherwise, the transmit power is increased ins steps until the maximum possible transmit power is reached. After that, if the packet is still not being received correctly, a random transmit power is chosen. The same principle can be followed by other nodes as well in order to achieve better performance of CI as demonstrated by DIPA.

VII. EVALUATION

In order to evaluate ReNEW module, we implemented it in Contiki OS [31] for WSNs based on our LWB implementation [32] and evaluated it on Indriya [15] testbed that offers realistic results. The experiments were conducted on 30 Tmote Sky nodes.

Algorithm 1 Slot Allocation Algorithm.

```
//txPower indicates the current transmit power
//txTime is the time required to complete one transmission
//We assume the power required for Tx and Rx are equal
//slotEnergy is the energy required to participate in a slot
At the beginning of the communication round $k$:
slotEnergy ← txPower * txTime * $\eta$ * 2;
$A(k)$ ← min($E(k)$, avg_harvested_energy($k$ - slotEnergy));
$n_{\text{slots}}$ ← $A(k)$/slotEnergy;
Sort the slots in descending order of their priority;
Schedule the first $n_{\text{slots}}$ for participation;
```

Algorithm 2 Transmit Power Adaptation Algorithm.

```
Function OnReceive (Packet p)
//txPower indicates the current transmit power
if p.IsCRCValid () == TRUE then
    Call decreaseTransmitPower ();
else
    if txPower == MAX_TRANSMIT_POWER then
        Call randomizeTransmitPower ();
    else
        Call increaseTransmitPower ();
    end if
end if
```

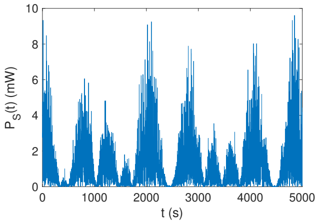
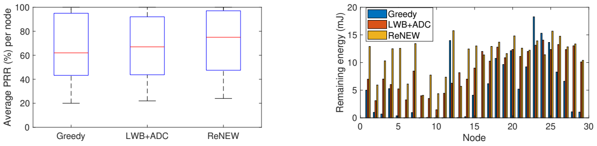


Fig. 6: Power trace with Moser's model



(a) (b)

Fig. 7: Scenario 30s, 20n and 50B: (a) Average PRR. (c) Remaining energy per node after 60 rounds.

A. *Evaluation Setup*

Energy modeling. We implemented the energy-harvesting battery model in software. That is, the amount of energy harvested is computed using an energy model (uniform or Moser's) on the nodes and added to the available energy at regular intervals. When an operation occurs, we subtract the energy consumed for that operation in the battery model. Since we use Contiki operating system, we use the *Energest module* to determine the energy consumed per operation. We consider that each node stores the harvested energy in a supercapacitor of size E_{max} =20 mJ. We performed experiments with a uniform arrival process having a mean rate of 50μJ/s. This is the average amount of energy that can be harvested from indoor lighting [29] which is significantly less than the amount of energy spent in a communication round. For example, a 100 B packet to be sent in an LWB slot with η=2 consumes almost 900μJ.

While the uniform distribution may correspond to the indoor lighting source, we also consider Moser's model that emulates a solar source [19]. The power harvested from this model is shown in Figure 6. The obtained power trace $P_S(t)$ exhibits both stochastic and deterministic with periodic behavior and simulates patterns of day and night periods similar to those experienced by solar cells in an outdoor environment. The evaluation with this model is done in the FlockLab testbed with 30 Tmote Sky nodes [16].

Application. The nodes need to report their sensed data periodically to a sink node. To evaluate ReNEW, we experimented with two communication round intervals of 30 s, and 60 s. For each interval, packets of different lengths (50 B and 100 B) and the different number of source nodes (10 and 20) are also experimented with. We chose these scenarios to test ReNEW for the potential worst-case scenarios.

In FlockLab, we evaluated 60 s intervals with different packet lengths (50 B and 100 B) and the different number of source nodes (10 and 20).

Algorithms. We compare ReNEW + LWB with (a) LWB with no energy-management algorithm. We call this a "greedy" energy allocation policy as the nodes try to participate in a slot if there is energy. (b) LWB with a well-known adaptive duty cycling technique [18] for EH-WSNs as the energy scheduler. The initial battery level is to 65% as considered in the paper. We denote this as "LWB+ADC". Note that all the algorithms employ forwarder selection and therefore participates *intelligently* in the necessary slots only.

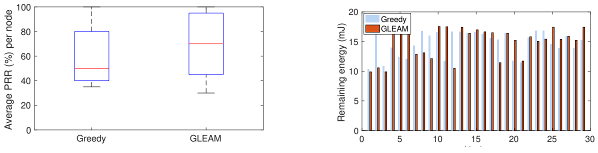
Metrics. The two metrics used are PRR and average remaining energy in the nodes to infer the lifetime indirectly. The PRR is measured at the sink node. Further, the sink node is considered to be connected to the power grid.

Realistic evaluation. The energy consumed on the nodes includes the energy spent on all aspects of the protocol including energy for the actual data collection, overheads for schedule distribution (also ACK in case of ReNEW), and retransmissions (η). Furthermore, the wireless channel conditions are uncontrolled and the experiments were conducted in the possible presence of WiFi and other interfering sources in the remote testbeds. Therefore, the results depict a real-world deployment scenario.

B. *Results*

A word on notation: In the figures, the 30 s and 60 s indicate the corresponding period of communication rounds, 30 seconds and 60 seconds, respectively; 10n and 20n indicate 10 and 20 source nodes that periodically send data (out of 30 nodes), respectively. The data size is either 50B or 100B (bytes).

At the outset, we set that all the nodes have a fully charged capacitor. Fig. 7(a) shows the average PRR of a 20 source node network, with a 30 s interval. Even though it is impossible to deliver all the packets with a low harvesting rate, it is clear that ReNEW improves the average PRR as opposed to both the other algorithms; in this case by at least 17% as compared



(a) (b)

Fig. 8: Scenario 60s, 20n and 50B with Moser's model: (a) Average PRR. (c) Remaining energy per node after 30 rounds.

to the greedy approach. The ADC mechanism is better than the greedy approach as it adapts to the available energy but is not better than ReNEW. This is due to the fact that when the nodes have higher energy, they behave similarly to the greedy approach. When the nodes have low energy, the nodes become conservative in their participation, leading to lower PRR.

Fig. 7(b) shows that the greedy approach drains almost all energy to maximize participation in slots whereas, ReNEW is more energy-aware. Thus, even if the harvesting rate drops in the next rounds, the network can sustain for a longer time. However, this does not affect (reduce) the PRR of the network as evident from Fig. 7(a).

Figure 8 shows the results from FlockLab testbed with Moser's energy arrival model. In Figure 8(a), we see that the average PRR from ReNEW is higher than that of the greedy LWB protocol. The median of ReNEW is around 20% higher than that of greedy LWB. Similar to the uniform distribution based energy arrival model, the average residual energy in every node is slightly higher with ReNEW than with greedy LWB as shown in Figure 8(b).

Heavy vs. Light traffic. Fig. 9(a) shows the PRR for data collection over 30s and 60s intervals sending 100 B of data. Evidently, with more time to harvest and lower the traffic, the performance of all the algorithms is almost similar. However, in the worst case (period being 30 s, and 20 source nodes), ReNEW shows that it can outperform by *2.5* times the greedy approach, and also LWB+ADC approach significantly. This performance is due to the multi-fold components of ReNEW, particularly dynamic node activation and power adaptation. Fig. 9(b) shows the light traffic scenario wherein 10 nodes transmit data and all the methods perform extremely well. Fig. 9(c) shows the average amount of energy remaining on the nodes for a payload length of 100 B. We see that ReNEW keeps a buffer of more energy on average. A big part of this is due to the dynamic node activation.

Figure 9(d) shows the average PRR for 60 s periodic data collection for a different number of source nodes and payload lengths, with Moser's energy arrival model. Similar to the results with the uniformly distributed energy arrival model, ReNEW has higher PRR in all cases. Figure 9(e) shows results with ReNEW achieving only slightly higher energy on average than greedy LWB in the scenario with 20 source nodes.

Payload length. The payload length also significantly influences the performance, as the larger the payload, the more is the required energy to transmit. Figure 9(a), Figure 9(b) and Figure 9(d) show the results when 50B and 100 B were sent by the source nodes for 60 s periodicity. It is again evident that more payload length has an influence on the performance. Again, ReNEW outperforms the other approaches.

Density. Fig. 9(a) clearly shows that the higher the density of redundant nodes, the better is the performance. Furthermore, due to the dynamic activation of redundant nodes, ReNEW performs better than the other approaches. In ReNEW, a node with energy less than 75% of its maximum capacity will choose with a probability, $p = 0.5$, to participate or not. This reduces the number of redundant nodes wasting their energy. As not all nodes exhaust energy in all participatable slots and due to this, there is a higher chance for ReNEW to find at least one forwarder to send its packets. This is future work as to how much this helps.

Latency. The average end-to-end latency, when the packets reached the sink, is tabulated in Table II. Evidently, the latency does not deviate from LWB or Glossy due to ReNEW.

Hop count	1	2	3	4	5
Delay (ms)	6	9	12	15	18

TABLE II: Average latency (rounded-off) obtained over different hops in the Indriya testbed.

C. Importance of Redundant Nodes

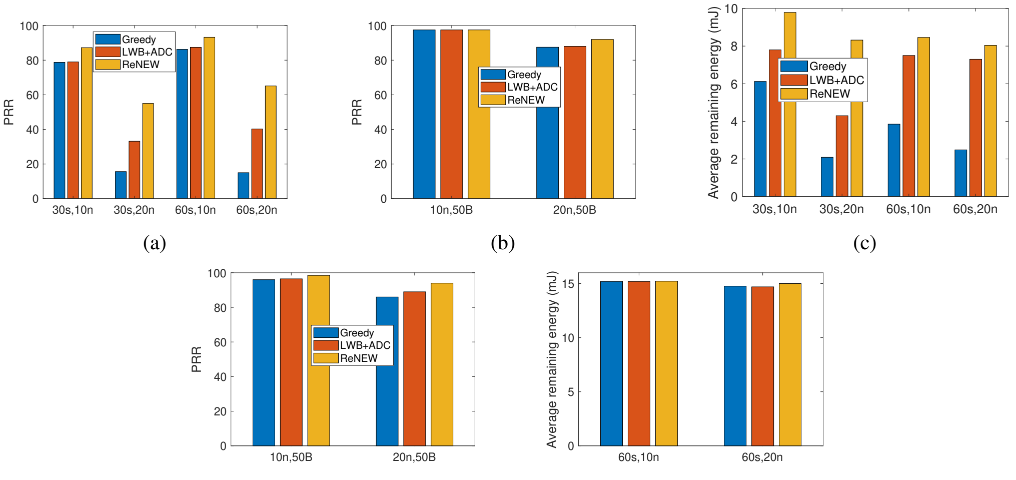
ReNEW outperforms greedy due to several factors described previously. However, ReNEW achieves a highly reliable data collection only when there is either high incoming energy or low traffic intensity. A third parameter, redundant nodes, also helps in boosting the reliability. We investigate the influence of these redundant nodes and the dynamic node activation in ReNEW.

In ReNEW, the nodes with the energy of more than 75% of their maximum capacity will participate in all its slots in a communication round. However, when it is below this number, the node chooses with a probability, $p = 0.5$, to participate or not. This reduces the number of redundant nodes wasting their energy unnecessarily. To evaluate the benefits, we implemented a dynamic node activation (DNA) for the greedy LWB scheme and compare them. Figure 10(a) shows the average PRR for different scenarios with 30 s communication round interval. We see that DNA is not helpful when there is light traffic, however, when the traffic intensity increases, the importance of redundant nodes also increases. There are more forwarders available for every transmitted packet for the DNA than the greedy approach. In fact, DNA helps the 2.5 times gain obtained in Figure 9(a), and further improved significantly by the other components.

An important observation can be made in Figure 10(c) that with increasing traffic intensity, while the average remaining energy decreases for the greedy approach, the remaining energy for DNA decreases only slightly. This indicates that many nodes were not activated always. Therefore, ReNEW outperforms the greedy approach significantly.

VIII. CONCLUSIONS

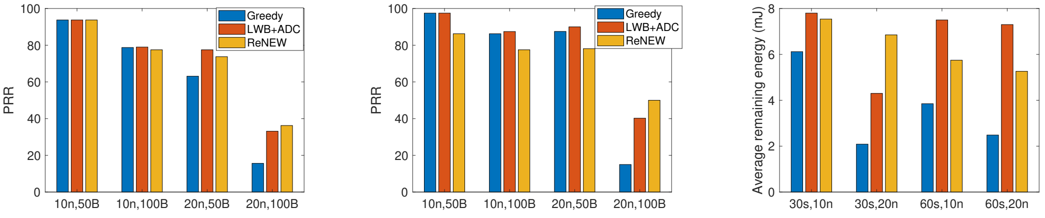
The Internet of Things (IoT) is changing our daily life bringing better and improved quality of life. However, these myriads of IoT devices powered by batteries cannot scale.



(a) (b) (c)

(d) (e)

Fig. 9: (a) Average PRR for different traffic intensities and source nodes (100 B). (b) Average PRR for payload length 50 B at 60 s periodicity. (c) Average remaining energy for a different number of sources, and payload length of 100 B. (d) Average PRR for periodicity of 60 s with Moser's model. (e) Average remaining energy for a payload length of 100 B with Moser's model.



(a) (b) (c)

Fig. 10: Evaluation of the dynamic node activation (DNA) scheme. (a) Average PRR for source nodes for 30 s interval. (b) Average PRR for source nodes for 60 s interval. (c) Average remaining energy for payload length of 100 B.

Thus, we sought ambient energy harvesting in WSNs to be used in IoT applications. However, these devices must provide a similar performance in terms of latency and reliability as their battery-powered counterparts.

In this paper, we focused on providing high reliability to EH-WSNs. We proposed to use a recent data collection protocol LWB based on Constructive Interference, which can provide guarantees on latency. However, in EH-WSNs setting LWB cannot guarantee reliability because of the stochastic nature of energy harvesting. To this end, we proposed a module called ReNEW. We proposed an *optimal policy* and also found the necessary condition for designing an optimal policy. We also show that it is indeed an optimal policy by showing the existence of a stationary (or a threshold vector). Furthermore, we proposed several enhancements and fine-tuned the protocol to improve the reliability offered by ReNEW. ReNEW is completely distributed and a practical module. We implemented ReNEW on TMote Sky nodes. We used Indriya and FlockLab testbeds, which are standard experimental facilities, to evaluate our algorithms. We found that ReNEW outperforms LWB even with an adaptive duty-cycling mechanism. A key reason for this performance improvement is the redundant nodes. Finding the critical density that can provide guarantees on reliability in EH-WSNs is an important challenge, which we will investigate in our future work.

ACKNOWLEDGMENT

InSecTT (www.insectt.eu) has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. Burns et al., "NIST: The IES-City Framework Draft Version 1.0," 2018, accessed: 30 July 2018. [Online]. Available: https://s3.amazonaws.com/nist-sgcps/smartcityframework/files/ies-city_framework/IES-CityFrameworkdraft_20180207.pdf

[2] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, 2014.

[3] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *2003 IEEE ICRA*, vol. 1. IEEE, 2003, pp. 19–24.

[4] Z. A. Eu, H.-P. Tan, and W. K. Seah, “Opportunistic routing in wireless sensor networks powered by ambient energy harvesting,” *Computer Networks*, vol. 54, no. 17, pp. 2943–2966, dec 2010.

[5] S. Yang, X. Yang, J. A. McCann, T. Zhang, G. Liu, and Z. Liu, “Distributed networking in autonomic solar powered wireless sensor networks,” *JSAC*, vol. 31, no. 12, pp. 750–761, 2013.

[6] C. Renner, S. Unterschütz, V. Turau, and K. Römer, “Perpetual Data Collection with Energy-Harvesting Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 11, no. 1, pp. 1–45, Sep. 2014.

[7] T. Roughgarden and É. Tardos, “How bad is selfish routing?” *Journal of the ACM (JACM)*, vol. 49, no. 2, pp. 236–259, 2002.

[8] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, “Low-power wireless bus,” in *ACM SenSys*. ACM, 2012, pp. 1–14.

[9] A. Escobar, C. Gonzalez, F. J. Cruz, J. Garcia-Jimenez, J. Klaue, and A. Corona, “Redfixhop: Efficient ultra-low-latency network flooding,” in *IEEE SECON*. IEEE, 2016, pp. 1–2.

[10] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *SenSys ’09*. ACM Press, Nov. 2009, p. 1.

[11] N. Burri, P. Von Rickenbach, and R. Wattenhofer, “Dozer: ultra-low power data gathering in sensor networks,” in *ACM/IEEE IPSN*, 2007.

[12] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, “Efficient network flooding and time synchronization with glossy,” in *ACM/IEEE IPSN*. IEEE, 2011, pp. 73–84.

[13] V. S. Rao, M. Koppal, R. V. Prasad, T. V. Prabhakar, C. Sarkar, and I. Niemegeers, “Murphy loves CI: unfolding and improving constructive interference in WSNs,” in *INFOCOM 2016*, apr 2016.

[14] V. Rao, R. Prasad, C. Sarkar, and I. Niemegeers, “Renew: A practical module for reliable routing in networks of energy-harvesting wireless sensors,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[15] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, “Indriya: A low-cost, 3d wireless sensor network testbed.” in *TridentCom*, vol. 90. Springer, 2011, pp. 302–316.

[16] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, “Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems,” in *ACM/IEEE IPSN*, 2013.

[17] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM TECS*, vol. 6, no. 4, sep 2007.

[18] C. M. Vigorito, D. Ganesan, and A. G. Barto, “Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks,” in *IEEE SECON*. IEEE, jun 2007, pp. 21–30.

[19] C. Moser, D. Brunelli, L. Thiele, and L. Benini, “Real-time scheduling for energy harvesting sensor nodes,” *Real-Time Systems*, vol. 37(3), 2007.

[20] N. Michelusi, K. Stamatiou, and M. Zorzi, “Transmission Policies for Energy Harvesting Sensors with Time-Correlated Energy Supply,” *IEEE TOC*, vol. 61, no. 7, pp. 2988–3001, jul 2013.

[21] J. Lei, R. Yates, and L. Greenstein, “A generic model for optimizing single-hop transmission policy of replenishable sensors,” *TWC*, vol. 8, no. 2, pp. 547–551, feb 2009.

[22] V. S. Rao, R. V. Prasad, and I. G. Niemegeers, “Optimal task scheduling policy in energy harvesting wireless sensor networks,” in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2015, pp. 1030–1035.

[23] H. Yang, Y. Ye, X. Chu, and M. Dong, “Resource and power allocation in swipt enabled device-to-device communications based on a non-linear energy harvesting model,” *IEEE Internet of Things Journal*, 2020.

[24] M. Dong, K. Ota, L. T. Yang, A. Liu, and M. Guo, “Lscd: A low-storage clone detection protocol for cyber-physical systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 5, pp. 712–723, 2016.

[25] D. Carlson, M. Chang, A. Terzis, Y. Chen, and O. Gnawali, “Forwarder selection in multi-transmitter networks,” in *2013 IEEE DCOSS*, 2013.

[26] K. Kar, A. Krishnamurthy, and N. Jaggi, “Dynamic node activation in networks of rechargeable sensors,” in *INFOCOM 2005*.

[27] P. Poggi, G. Notton, M. Muselli, and A. Louche, “Stochastic Study of Hourly Total Solar Radiation in Corsica Using a Markov Model,” *International Journal of Climatology*, vol. 20(14), 2000.

[28] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, “Optimal energy management policies for energy harvesting sensor nodes,” *arXiv preprint arXiv:0809.3908*, 2008.

[29] M. Gorlatova, M. Zapas, E. Xu, M. Bahlke, I. J. Kymissis, and G. Zussman, “CRAWDAD dataset columbia/enhants (v. 2011-04-07),” [Online] http://crawdad.org/columbia/enhants/20110407/energy.

[30] M. R. Garey and D. S. Johnson, *Computers and intractability*. wh freeman New York, 2002, vol. 29.

[31] A. Dunkels, B. Grönvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” in *LCN 2004*. IEEE, 2004, pp. 455–462.

[32] C. Sarkar, “Lwb and fs-lwb implementation for sky nodes using contiki,” arXiv preprint - https://arxiv.org/pdf/1607.06622.pdf, 2016.