Sink's One-Hop Neighborhood Energy Hole Mitigation Scheme for Dense Wireless Sensor Networks

Gokou Hervé Fabrice Diédié [1,*], Armand Kodjo Atiampo [2], and Tchimou N'Takpé [3]

[1] Laboratory of Mathematics and Computer Science, Université Peleforo Gon Coulibaly, Korhogo, Ivory Coast
[2] Unité de Recherche et d'Expertise du Numérique, Université Virtuelle de Côte d'Ivoire, Abidjan, Ivory Coast;
Email: armand.atiampo@uvci.edu.ci (A.K.A.)
[3] Laboratory of Mathematics and Computer Science, Université Nangui Abrogoua, Abidjan, Ivory Coast;
Email: tchimou.ntakpe@gmail.com (T.N.)
*Correspondence: herve.diedie@upgc.edu.ci (G.H.F.D.)

***Abstract*—In Wireless Sensor Networks (WSNs), nodes close to the sink forward traffic from all over the network. Therefore, they tend to deplete their battery and be congested more than the other sensor nodes. Addressing this issue known as the hot spot problem is critical to design an efficient network. In this paper, we formulated this question as a combination of the precedence-constrained knapsack and the single-runway aircraft landing problems. The resulting TDMA (Time Division Multiple Access)-based protocol leverages two mixed-integer linear programs to select nodes and schedule data transmissions. This scheme simultaneously considers parameters like energy level, congestion degree, link state, transmission delay, and packet priority using a single queue, unlike solutions commonly found in the literature. Simulation results showed that this protocol can improve packet delivery ratio, throughput, fairness, and network lifetime respectively by 5.06%, 35.20%, 32.49%, and 60% compared with that of IEEE 802.14.5-based counterparts.**

***Keywords*—hot spot, queue management, packet prioritization, TDMA, wireless sensor networks**

I. INTRODUCTION

A typical Wireless Sensor Network (WSN) consists of a large number of small sensing devices and a gateway also called the sink. WSNs have a wide range of applications in domains such as environment, health, industry, agriculture, transportation [1–4]. In this kind of networks, sensor nodes are in charge of data collection and routing to the gateway [5, 6]. In this many-to-one communication pattern, the sink's one-hop neighbors are the most solicited sensor nodes. Therefore, they tend to drain their energy faster than others; leading to in the short term (quickly) to an isolation of the sink. This major issue is known as the hot-spot or energy hole problem [7, 8].

A common solution involves scheduling transmissions so as to prevent collisions and congestions. This strategy is usually coupled with different routing and data aggregation schemes [9–11]. They are generally executed at the application, routing, or MAC (Medium Access Control) layer. Most of such contributions found in the literature apply these strategies in the entire network. However, several studies have shown that the vicinity of sink requires a specific policy to really cope with local energy holes. In this paper, we propose a TDMA-based solution that considers both nodes' congestion degree and link state in order to mitigate energy consumption inherent to retransmissions due to packet losses. We also applied a packet prioritization scheme to better adapt to most real-world scenarios. The resulting protocol helps minimize packet losses, enhance throughput and prolong network lifetime. The main contributions of this work can be summarized as follows:

(1) A formulation of the packet selection process as a combination of the precedence-constrained knapsack and the single runway aircraft landing problems;
(2) A genetic algorithm-based heuristic to select senders according to link state, packets' priority and sink node's capacity;
(3) A specific heuristic to create a TDMA-based transmission non-preemptive scheduling using time slots with dynamic size considering packet priority from a single queue.

The rest of the paper is organized as follows: Section II reviews the related contributions. The solution proposed is detailed in Section III. Experiments, their results and discussions are presented in Section IV. The conclusion is provided in Section V.

II. RELATED WORK

In the past two decades, numerous solutions have been designed to address the hot spot problem. They are generally categorized according to the layer where they are executed (MAC, Routing, Transport, or Application).

Manuscript received June 11, 2023; revised August 5, 2023, accepted August 22, 2023.

However, solutions specifically applied to the sink node's vicinity are usually MAC layer-oriented. They are classified into three categories: contention-based protocols, scheduled-based also called contention-free protocols, and hybrid ones [12‒15].

In the first category, solutions are mainly based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. They use different schemes whether synchronous or asynchronous. In WSNs, CSMA/CA mechanisms are broadly applied through the IEEE 802.15.4 standard. Two schemes are generally used, namely: the slotted and the unslotted mechanisms; where beacons are respectively used and avoided for the synchronization of nodes. The slotted CSMA/CA is generally preferred in WSNs [16]. However, under high-traffic scenarios, this mode is more energy-consuming and has a lower throughput [17, 18]; while the number of the reserved time slots is limited to seven. Many solutions have been proposed to cope with such shortcomings, especially by adjusting the size of the contention window [19]. Nevertheless, most of these solutions do not consider packet prioritization and thus are limited to only some time-driven applications. Several recent works are aimed to overcome this limitation.

Onwuegbuzie *et al*. [20] proposed the concept of Class of Service (CS) i.e., a service differentiation technique used to handle heterogeneous data or traffic. Services are assigned unique levels of priority according to their data or traffic. This scheme can be applied in both single-hop and multi-hop scenarios. The traditional MAC frame format of the IEEE 802.15.4 was modified to accommodate data prioritization, with an improved beacon called Beacon Plus Plus (B++). The binary exponential backoff parameters were also modified. However, this solution does not consider the destination node's capacity or the transmission delay. By contrast, Bouazzi *et al.* [21] designed a strategy to arrange nodes into two priority groups based on their traffic rate and the transmission delay. Besides, they suggested dynamically modifying the length of the Contention Access Period (CAP) duration in order to improve the QoS. However, the priority is actually assigned to nodes that have high traffic rate and random transmission delay. This strategy raises the number of packet losses. Thus, can hardly be applied specifically to the sink's neighborhood; especially, in rare but critical event-driven applications.

Huamei *et al*. [22] suggested a wake-up matching mechanism initiated by the receiver based on a multi-priority back-off. The sender adjusts its schedule according to the wake-up information of the destination node. The sender then calculates the receiver's next active time and adjusts it to its own. Priority level is calculated leveraging the remaining energy of nodes and their queue length. Unfortunately, this strategy increases the end-to-end delay; since when a node needs to send data, it has to wait for the receiver's next wake-up time.

Gonzàlez *et al.* [23] introduced the concept of a virtual sink formed by the sink and its one-hop neighbors called satellites. They proposed a strategy consisting of three phases. The bandwidth of the virtual sink is expanded to reduce packet loss in this area. Slot allocation is assigned to the sink node. The latter gathers information from its neighbors about their queue length and incoming traffic rate. Then it defines priorities based on each satellite's demand. Regrettably, this strategy is not scalable since the virtual sink supports up to eight nodes (i.e., the sink and seven satellites).

Nguyen *et al*. [24] used a similar strategy dedicated to Sink's one-hop neighborhood. When a node needs to transmit a packet, it randomly sends an RTS (Request to Send) message to the sink in the contention window. Only the nodes that will receive a CTS (Clear to Send) message remain active. Contention window is split according to traffic proportions. These can be defined based on the packets' priority level. Unfortunately, in this solution, only one RTS is accepted in each cycle. Therefore, the delay could increase with the number of nodes. This scheme may not scale with the number of traffic categories. In addition, the selection process is not realistic since the nodes' residual energy is not considered.

Sakib *et al*. [11] presented a double scheduling scheme where nodes start listening to the channel as soon as they wake up. After timer expiration, all the nodes broadcast a synchronization message. The sensor nodes adopt the two schedules if the sink sends a schedule that differs from theirs. Once the synchronization is done, the sink waits for data transmissions until the waiting timer expires. Any transmission session can be pre-empted if the sink receives a request with higher priority from a sensor that differs from the current sender. The sink will wait for the intended packet after allowing the new transmission then shifts to sleep state at the expiration of the timer. This solution is extensible to any number of priority levels, but the pre-emption technique is detrimental to fairness and can increase packet losses when facing concurrent high priority demands.

The second category of MAC layer-based solutions for the hot spot problem consists of contention-free protocols. They leverage methods that build transmission schedules to prevent collisions. These solutions are generally divided into Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Time Division Multiple Access (TDMA) protocols. In other words, they try to prevent congestion and collisions by respectively allocating a different frequency band, code sequence, and time slot to each node. In WSNs, current FDMA solutions are not recommended due to a narrow bandwidth [25]. Most of the contention-free protocols commonly found in the literature use TDMA [26]. These protocols can optimize the bandwidth [27]. Note that, finding a schedule that minimizes the number of time slots to achieve convergecast is believed to be NP-hard [28]. This problem is also called TDMA Frame Minimization Problem [29].

One of the earliest TDMA-oriented solutions for the hot spot problem is LEACH (Low-Energy Adaptive Clustering Hierarchy) proposed by Heinzelman *et al.* [30]. In the past two decades, it has inspired numerous protocols. Regrettably, most of them have not considered packet prioritization [31–33]. However, few recent solutions have tried to address this issue.

Kaur and Kumar [34] designed a cluster-based protocol that prioritizes packets of sensor based on their residual energy and queue length. Unfortunately like most LEACH-based schemes, the proposed protocol uses TDMA-oriented schedules only for intra-cluster transmissions. Communications between Cluster Heads (CHs) and the sink are based on traditional CSMA/CA techniques. This strategy could not be scalable. The same authors proposed a similar approach using a fuzzy logic-based but intra-cluster scheduling [35].

Muzakkari *et al.* [36] suggested a scheme that considers both nodes' queue length and priority level of packets. To ensure fairness i.e., solve the problem of starvation suffered by data with low priority, they propose using an exponential weighted moving average. Each node can extend the duration of its active period when traffic rate increases. A sender must inform a receiver about the duration of its duty-cycle at the beginning of data transmission in order to have their active periods be synchronized. Although it can be applied in the sink's vicinity, this implicit TDMA scheme was not specifically dedicated to this area. Therefore, synchronization process between the sink and its neighbors could hardly scale.

Jeon and Park [37] proposed a priority mechanism aimed to explicitly cope with fairness. They introduced the concept of weighted scheduling. Data are grouped by a relay node according to their destination and priority. Indeed, each node creates temporary buffers, to rearrange data according to their destinations; then uses different buffers respectively for each level of priority (referred to as weight). This strategy guarantees transmission of data even with very low-priority but requires the creation of one buffer per weight. In addition, this scheme can hardly be applied to resource-constrained devices like wireless sensors. Moreover, the proposed protocol does not consider the capacity of the receiver (the relay node or the sink).

A few hybrids priority-aware MAC solutions exist in the literature. Recently, Lin *et al*. [38] suggested a cluster-based scheme that combines CSMA and TDMA techniques respectively for transmission scheduling and to coordinate the priority-aware transmission, which allows radio channel reuse by multiply rescheduling to different priority classes of packets. Unfortunately, this strategy is not scalable and is not specifically meant for the sink's vicinity.

Poudel and Moh [39] suggested a cluster-based solution for UAV (Unmanned Aerial Vehicle)-aided WSNs; where the UAV plays the role of a mobile sink. This solution dynamically adapts the number or the size of slots to the varying amount of data traffic. The scheduling phase begins after UAV receives registration frames from the sensors it covers. Afterward, the UAV assigns priority to CHs based on their remaining energy and the amount of data they need to transmit. Time slots are allocated based on that priority. The lengths of these slots are decided considering the size of the CHs' buffer. After collecting data from the cluster members, CH sleeps and periodically wakes up to check beacon signals. Once CHs receive beacon signals from the UAV, they get active and immediately contend for sending registration frames to the UAV. During this phase, CHs use CSMA and a random back-off scheme in case of collision. Regrettably, such frequent collisions result in energy waste and packet losses.

III. Proposed Method

This section aims to present our solution. We first discuss our motivations, objectives, and assumptions, then detail our scheme, referred to as Last Mile Delivery Protocol-LMDP. This protocol is distributed and asynchronous.

A. Motivation and Objectives

As shown in the previous section, only a few papers have addressed the hot spot problem, specifically in the sink's vicinity. Moreover, when traffic is high, TDMA-oriented solutions are generally preferred to prevent energy waste and time slots limitations [17, 18]. Additionally, to be realistic, the proposed scheme must consider packet prioritization.

Regrettably, to the best of our knowledge, so far, no TDMA-oriented and priority-aware scheme has been specifically dedicated to the sink's one-hop neighborhood.

This work is aimed to address this issue while providing a good trade-off between network efficiency and lifetime irrespective of the number of priority levels.

B. Assumptions

We assume that:

- Nodes are equipped with an omni-directional radio;
- Each node has a unique identifier (ID);
- Nodes are uniformly and randomly deployed in the area of interest;
- Nodes' connection is modeled as an UDG (Unit Disk Graph);
- Each node can assess distances through the received signal strength or a specific localization protocol;
- Each node has a finite buffer that helps to queue data packets with a FIFO (First-In First-Out) policy.

C. Description

Let $G = (V; E)$ be the graph induced by the sink's neighborhood in a WSN. $V$and E respectively denote the set of nodes (i.e., the sink and its neighbors) and their links.

With $E = \{(u, v) \in V \times V : d(u, v) < (r_u + r_v)\}$ where $d(u, v)$ is the Euclidean distance between nodes u and v;

Note that r_u and r_v respectively denote their communication ranges. Fig. 1 depicts a sink's one-hop neighbourhood.

LMDP uses a message-passing communication model.

After neighbor discovery, its operation is divided into rounds. At the beginning of each round, the sink s

broadcasts a DATA-REQ message to its one-hop symmetric neighbors and triggers a timer (*DATA-timer*) then waits for any response during $2 \times t_{wait}$ seconds. This duration is calculated using Eq. (1); where rtt(.) denotes the round-trip-time (in seconds) experienced with sensor node u during the neighbor discovery phase. N refers to the neighborhood of sink s.

$$t_{wait} = \max\{\text{rtt}(u); u \in N\} \quad (1)$$

The DATA-REQ message contains its ID and the *duty size* (i.e., residual space) of its buffer denoted by L .

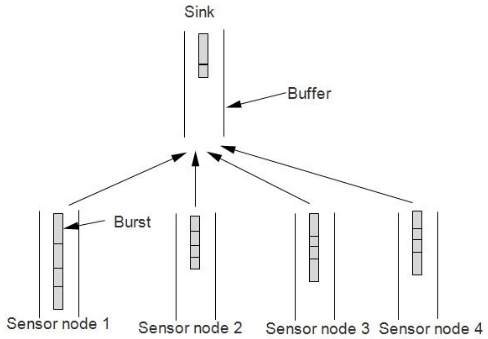

Figure 1. Example of a sink's one-hop neighborhood. Sensor nodes contend for sending their bursts.

On receiving such a message, a sensor node prepares a *burst* of data packets that matches the sink's demand (see Fig. 1); then replies by sending back a DATA-ACK message. The latter contains its ID, the priority index, and the size of each data packet to be sent.

Note that, when a sensor node has no data to send, it must discard the received DATA-REQ.

When receiving a DATA-ACK message from a neighbor, the sink estimates the quality of the link from this node via two metrics namely, SNR (Signal to Noise Ratio) and SINR (Signal Interference plus Noise Ratio).

After the *DATA-timer* expiration, the sink must determine which bursts of data packets to simultaneously receive according to its buffer's *duty size*, its residual energy, and the duration of the current round (i.e., its duty-cycle). We address this issue as a variation of the well-known Precedence-Constrained Knapsack Problem [40, 41] formulated via the integer linear program expressed by Eq. (3)–Eq. (11).

So let:

N be the set of neighbors;

B be the number of bursts proposed by neighbors;

$$x_{ij} = \begin{cases} 1, \text{if neighbor } i'\text{s burst } j \text{ is picked} \\ 0, \text{otherwise} \end{cases}$$

ℓ_{ij} be the length of neighbor *i*'s burst *j*;

ξ_{ij} be the energy wasted handling neighbor *i*'s burst *j*;

δ_{ij} be the transmission delay of neighbor *i*'s burst *j*;

Υ_{ij} be the *relative priority* of neighbor *i*'s burst *j*;

Er be the sink's residual energy;

R be the duration of current round;

R is calculated using Eq. (2); where η denotes the size of a data packet and L denotes current capacity of the sink's buffer;

$$R = t_{wait} \times \frac{L}{\eta} \quad (2)$$

Note that:
$\forall i \in N, \forall j \in B; (\ell_{ij} > 0) \wedge (\xi_{ij} > 0) \wedge (\delta_{ij} > 0)$

$$\max \sum_{i\in N} \sum_{j\in B} \Upsilon_{ij} x_{ij} \quad (3)$$

st:

$$\sum_{i\in N} \sum_{j\in B} \ell_{ij} x_{ij} \leq L \quad (4)$$

$$\sum_{i\in N} \sum_{j\in B} \xi_{ij} x_{ij} \leq Er \quad (5)$$

$$\sum_{i\in N} \sum_{j\in B} \delta_{ij} x_{ij} \leq R \quad (6)$$

$$x_{ij} - x_{ik} \geq 0; \forall i \in N, \forall j, k \in B, k > j \quad (7)$$

$$\ell_{ij} \leq L; \forall i \in N, \forall j \in B \quad (8)$$

$$\xi_{ij} \leq Er; \forall i \in N, \forall j \in B \quad (9)$$

$$\delta_{ij} \leq R; \forall i \in N, \forall j \in B \quad (10)$$

$$\Upsilon_{ij} > 0; \forall i \in N, \forall j \in B \quad (11)$$

Eq. (3) refers to the objective, namely maximizing the total priority index of the proposed bursts during this round. Eq. (4) ensures that the total length of the chosen bursts does not exceed the buffer's *duty size*. Eq. (5) states that the whole energy loss should not exceed residual energy. Eq. (6) guarantees that the whole transmission delay does not exceed this round. Eq. (7) requires that bursts' queuing order is respected. Eq. (8) asserts that no burst can overflow the buffer. Eq. (9) states that the sink's residual energy must suffice to handle (i.e., to receive and transfer) any of the proposed bursts. Eq. (10) requires that each proposed burst has a fitness score. Eq. (11) suggests that each burst needs to be transferred.

This program is useful if all the proposed burst cannot be picked, i.e., $\left(\sum_{i\in N} \sum_{j\in B} \ell_{ij} > L\right) \vee \left(\sum_{i\in N} \sum_{j\in B} \xi_{ij} > Er\right) \vee \left(\sum_{i\in N} \sum_{j\in B} \delta_{ij} > R\right)$. Such a situation is very common in dense Wireless Sensor Networks.

ξ_{ij} is calculated using Eq. (12) where ERX_{ij} and ETX_{ij} respectively denote the energy wasted when receiving and forwarding neighbor *i*'s burst *j*; both ERX_{ij} and ETX_{ij} are calculated via the underlying energy consumption model.

$$\xi_{ij} = ERX_{ij} + ETX_{ij} \quad (12)$$

We leverage the SNR (Signal to Noise Ratio) and SINR (Signal Interference to Noise Ratio) to estimate bursts' transmission delays. These parameters are two well-known link quality indicators.

Eq. (13) and Eq. (14) help estimate δ_{ij}.

$$\delta_{ij} = \frac{\ell_{ij}}{C_{ij}} \quad (13)$$

$$C_{ij} = W_{ij} \times \log_2(1 + SINR_{ij}) \quad (14)$$

Note that C_{ij} is the *interfered capacity* (i.e. maximum transmission rate interferences) of the link that outgoes from neighbor *i* when sending burst *j*; where W_{ij} and

$SINR_{ij}$ respectively denote the bandwidth and the SINR of this link.

$$\tilde{\delta}_{ij} = \frac{\ell_{ij}}{\tilde{C}_{ij}} \tag{15}$$

$$\tilde{C}_{ij} = W_{ij} \times \log_2(1 + SNR_{ij}) \tag{16}$$

Eq. (15) and (16) estimate the ideal delay $\tilde{\delta}_{ij}$ for neighbor *i*'s burst *j*; while $\tilde{C}_{ij}$ denotes the *ideal capacity* of the link from the same neighbor; SNR_{ij} is the SNR of this link.

$$\Upsilon_{ij} = \begin{cases} \Upsilon_{ij-1} + \tilde{\Upsilon}_{ij}, \text{if } j \neq |N| \\ \tilde{\Upsilon}_{ij}, \text{otherwise} \end{cases} \tag{17}$$

Υ_{ij} is estimated using Eq. (17); where $\tilde{\Upsilon}_{ij}$ denotes the *absolute priority* assigned by neighbor *i* to its burst *j*.

$\tilde{\Upsilon}_{ij}$ in turn is calculated using Eq. (18), where $P(ij)$ and $\theta_k \in \mathbb{N}^*$ respectively denote the set of data packets in neighbor *i*'s burst *j* and the priority index that *i* assigned to data packet *k*.
Note that without loss of generality, we have $\theta_k \in [0,1]$.

$$\tilde{\Upsilon}_{ij} = \max_{k \in P(ij)} \{\theta_k\} \tag{18}$$

It is noteworthy that the packet prioritization policy is left to the underlying application. For instance, nodes could assign a priority index to packets according to their sojourn times. If so, one may consider that $\theta_k = \left\lceil \frac{w_k}{\hat{w}} \times \hat{\theta} \right\rceil$; where $w_k, \hat{w}, \hat{\theta}$ respectively denote packet *k*'s sojourn time, the maximum authorized sojourn time (i.e., waiting time threshold) and the maximum priority index value.

Regrettably, the 0-1 Knapsack Problem (KP) was proven NP-Complete [42]. Nevertheless, this problem or its variants can be solved using numerous efficient state-of-the-art heuristics. We propose using a scheme based on Genetic Algorithm (GA) [43]. The latter metaheuristics is deemed to efficiently find feasible solutions for 0-1 KP [44–46].

We encode chromosomes (i.e., solutions) leveraging matrix x. Therefore, x_{ij} can be referred to as the gene *j* brought by *i*.

We use roulette-wheel selection. Then we apply this classical scheme:

Step1) randomly generate a population of chromosomes;
Step 2) calculate the fitness score of all chromosomes;
Step 3) create a new population:
 Step 3.1) select the 2 fittest chromosomes;
 Step 3.2) perform crossover on these 2 chromosomes;
 Step 3.3) mutate the resulting chromosomes;
Step 4) replace the current population by the new one;
Step 5) if number of iterations is reached then stop. Otherwise, return the best solution in the current population and go to Step 2.

It is noteworthy that when generating population, we consider the hardest constraint i.e., the burst precedence as stated by Eq. (6).

After choosing the different bursts to be sent, the sink must define a TDMA-like schedule for the concerned neighbors. This schedule is aimed at allocating a transmission time slot to each neighbor so that the round timeline is not exceeded. We address this second issue as a variation of the well-known Aircraft Landing scheduling Problem (ALP) at an airport with a single runway [47, 48]. We formulate it using a mixed integer linear program via Eq. (19)–Eq. (22).

So let:
n be the number of the chosen bursts.

$$y_{ij} = \begin{cases} 1, \text{if the } i^{th} \text{ burst is received before the } j^{th} \text{one} \\ 0, \text{otherwise} \end{cases}$$

Ub_i be the arrival time window upper bound of *i*th burst;
Lb_i be the arrival time window lower bound of *i*th burst;
$\tilde{t}_i$ be the target (i.e., preferred) arrival time of *i*th burst;
t_i be the arrival time of *i*th burst;
Δt_i be the lag on arrival of the *i*th burst;
ψ_{ij} be the delay between arrivals of *i*th and *j*th bursts;
ρ_i be the delay penalty imposed to burst *i*;

$$\min \quad \sum_i^n \Delta t_i \times \rho_i \tag{19}$$

st:

$$y_{ij} + y_{ji} = 1, \forall i,j \in \{1,\dots,n\}: i \neq j \tag{20}$$

$$t_i + \psi_{ji} - (\Omega \times y_{ij}) = 1, \forall i,j \in \{1,\dots,n\}: i \neq j \tag{21}$$

$$0 \leq \Delta t_i \leq Ub_i - \tilde{t}_i, \forall i \in \{1,\dots,n\} \tag{22}$$

Note that $\Omega = Ub_i + \psi_{ij} - Lb_j$ and $\Delta t_i = t_i - \tilde{t}_i$

Eq. (19) states the objective i.e., minimize the total delay (lag) of the bursts to be received. Eq. (20) ensures that the chronological order of arrivals is linear. Eq. (21) ensures that arrivals times are separated, to prevent collisions. Eq. (22) requires that arrivals times remain in the defined window.

Ub_i and Lb_i are obtained using Eq. (23) and Eq. (24) where ΔR is the time elapsed since the beginning of the current round.

$$Lb_i = \Delta R \tag{23}$$

$$Ub_i = \Delta R + \max(\delta_{ix}; \tilde{\delta}_{ix}) \tag{24}$$

Note that δ_{ix} and $\tilde{\delta}_{ix}$ are obtained respectively via Eq. (13)–Eq. (15) for a sender *x*.

$\tilde{t}_i$ without loss of generality, is chosen as the middle of each time window $[Lb_i; Ub_i]$. ψ_{ij} is actually the average sojourn time in the sink's buffer estimated using Eq. (25); where λ and ϕ respectively denote the average arrival rate and the mean number of packets in this buffer.

$$\psi_{ij} = \frac{\phi}{\lambda} \tag{25}$$

λ and ϕ are periodically measured by the sink leveraging both incoming and outgoing traffic flows.

Eq. (26) helps calculate the penalty per unit of time ρ_i; where Υ_{xi} as a reminder, denotes the *relative priority* of burst *i* assigned by neighbor *x* (see Eq. (17));

$$\rho_i = e^{\Upsilon_{xi}}, x \in N \tag{26}$$

The scheme is to take a *target schedule* (i.e., a feasible arrival sequence) and compute an optimal one that minimizes delays.

Fig. 2 depicts a round from the sink's point view. Each round consists of time slots of different sizes; since time slots vary with the size of the burst to be sent. At the beginning of a round, the sink selects the *bursts* of data packets provided by its neighbors. This selection phase must consider the capacity of the sink's buffer.

Figure 2. Timeline showing LMDP operation.

Fig. 3 illustrates a schedule from the sink's point of view. The scheduling phase begins just after the selection phase. The upper and lower bounds of a time window are defined according to the estimated transmission delays respectively for both the best and the worst scenarios (i.e., with and without interference).

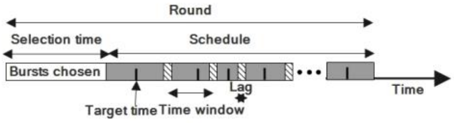

Figure 3. Timeline showing a schedule from the sink's perspective.

Fig. 4 depicts a round and a schedule from the sink's neighbors' point of view. This schedule defines transition delays between transmission and sleep (i.e., off-duty) states.

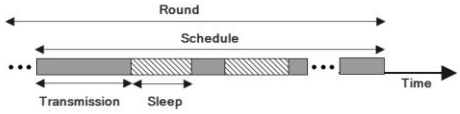

Figure 4. Timeline showing a schedule from a neighbor's perspective.

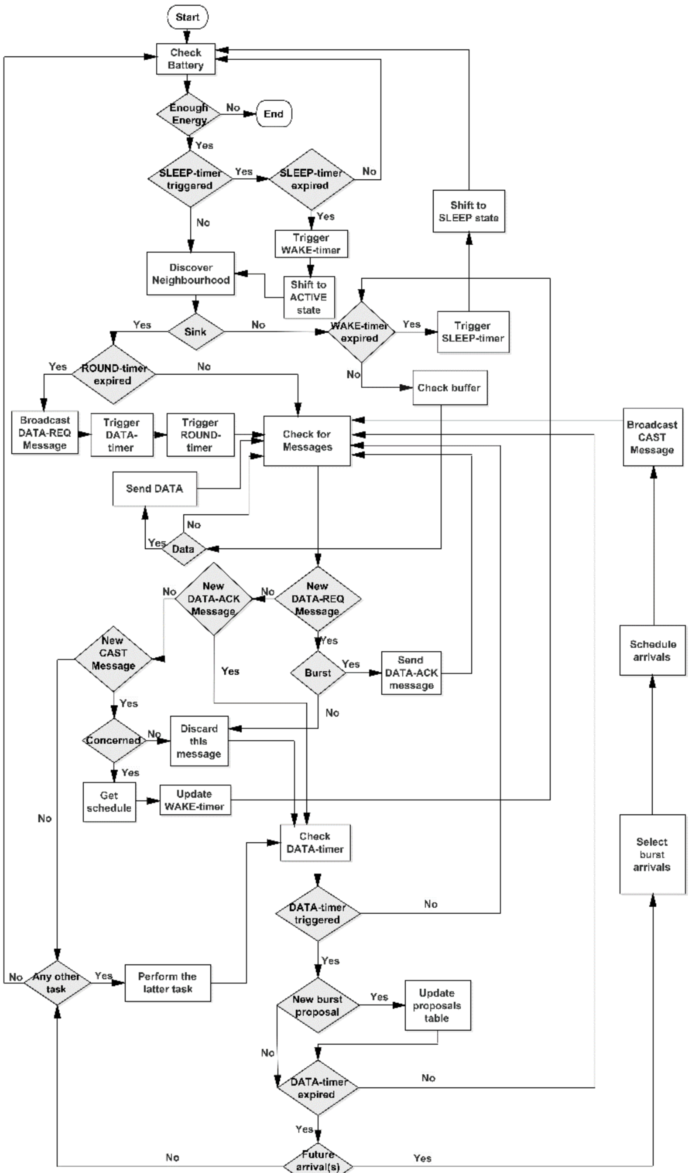

Figure 5. Flowchart of LMDP.

The rationale behind our solution is described in Fig. 5

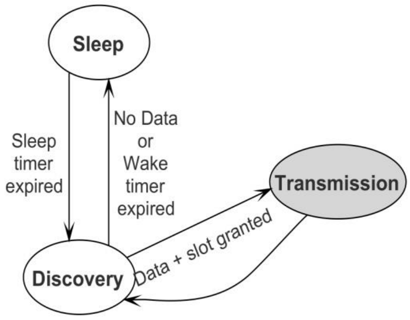

Figure 6. Duty-cycle of a neighbor inside a round.

Fig. 6 shows the duty-cycle of any sink's neighbor during a round. Such a node remains in sleep state until its sleep timer is expired. Then, it wakes up and enters into discovery state to check if it has data to send and whether the sink can receive them. If so, it sends these data until the assigned time slot is over.

Note that neighbors that have not been chosen remain in sleep state all along the round. It returns to sleep state if it has no data to send or its duty time is over.

The suggested model aims to make the durations of neighbors' active state and transmission state coincide. Unfortunately, ALP (Aircraft Landing Problem) has also been proven NP-hard [47, 48]. Several heuristics exist in the literature to solve this problem [49–51]. We propose using the scheme described in Algorithm 1.

Algorithm 1 Schedule construction

Input:
$N, B, x, \Upsilon, \Delta R, Lb, Ub, \delta, \tilde{\delta}, t, \tilde{t}, \psi, \Delta t, \rho, itmax, S^*$
Output: S^*
$B' \leftarrow \{(i;j) \in N \times B \mid x_{ij} = 1\}$
Calculate $\Upsilon_{ij}, \forall (i;j) \in B'$ see Eq. (17)
$B \leftarrow$ Sort B' according to Υ_{ij}
$f^* \leftarrow +\infty$
$S^* \leftarrow \emptyset$
$iter \leftarrow 1$
while $iter \leq itmax$ **do**
$f \leftarrow 0$
$S \leftarrow \emptyset$
for each $\alpha \in B' : i, j \in \alpha$
$Lb_\alpha \leftarrow \Delta R$
$Ub_\alpha \leftarrow Lb_\alpha + \max\{\delta_{ij}; \tilde{\delta}_{ij}\}$
$\tilde{t}_\alpha \leftarrow Lb_\alpha + \frac{\max\{\delta_{ij}; \tilde{\delta}_{ij}\}}{2}$
$t_\alpha \leftarrow \text{random}(\tilde{t}_\alpha; Ub_\alpha)$
$\Delta R \leftarrow \begin{cases} Ub_\alpha, \text{if } \alpha > 1 \\ Ub_\alpha + \psi_{\alpha\alpha-1}, \text{otherwise} \end{cases}$
$\Delta t_\alpha \leftarrow t_\alpha - \tilde{t}_\alpha$
$\rho_\alpha \leftarrow e^{\Upsilon_{ij}}$
$f \leftarrow f + (\Delta t_\alpha \times \rho_\alpha)$
$S \leftarrow S \cup \{(Lb_\alpha; Ub_\alpha; i; j)\}$
end for
if $f < f^*$ **then**
$f^* \leftarrow f$
$S^* \leftarrow S$
end if
$iter \leftarrow iter + 1$
end while

IV. Experiment Setup

In this section, we describe the simulation campaign we carried out. All the experiments were conducted using OMNeT++ simulator version 6.0 [52]. The parameters we used are summarized in Table I to Table V.

TABLE I. Simulation General Parameters

Parameter	Value
deployment zone	200 m × 200 m
number of sensor nodes	50 –200
location of the sink	(100;100)
sensor nodes' transmission ranges	135 m
sink's transmission range	150 m
sensor nodes' initial energy	2.5 J
sinks' initial energy	20 J
self-discharge per second	0.1 μJ
E_{elec} (see [30])	50 nJ/bit
e_{fs} (see [30])	10 pJ/bit/m^2
e_{amp} (see [30])	0.0013 pJ/bit/m^4
d_0 (see [30])	87 m
length of a typical data packet	512 bytes
links' bandwidth	2.4 GHz
links' bit rate	250 Kb/s
buffer queuing model	M/M/1
sensors' buffer arrival rate	[50,200] Kb/s
sink's buffer departure rate	[20,250] Kb/s
buffer size	1000 packets
priority index	[1,10]
packet drop delay	60 ms
maximum number of iterations	200

Table III presents parameters we used to randomly and uniformly vary link quality: namely, PRR (Packet Reception Ratio), SNR (Signal-to-Noise Ratio), SINR (Signal-Interference-plus-Noise Ratio), and LQI (Link Quality Indicator) [53].

Tables II and IV are inspired by the specifications of the IEEE 802.15.4/CC2420 standard [54].

TABLE II. State Transition Delay

	R_x (μs)	T_x (μs)	Sleep (μs)	Idle
R_x	–	1	194	–
T_x	1	–	194	–
Sleep	5	5	–	–
Idle	–	–	–	–

TABLE III. Link Quality Parameters

	PRR	SNR (dBm)	SINR(dBm)	LQI
Excellent	1	[40, 60]	[30, 40]	[106, 255]
Good	[0.75, 1]	[25, 40]	[15, 30]	[102, 106]
Medium	[0.35, 0.75]	[15, 25]	[5, 15]	[80, 102]
Poor	[0, 0.35]	[0, 15]	[0, 5]	[0, 80]

Note that the maximum bit rate of links was set to 250 Kb/s inspired by the IEEE 802.15.4 standard [54, 55].

For transmission energy dissipation we used the model proposed by Heinzelman *et al*. [30];

TABLE IV. POWER DISSIPATION OF STATE TRANSITION

	R_x (mW)	**T_x (mW)**	**Sleep (mW)**	**Idle**	
R_x	–	62	62	–	
T_x	62	–	62	–	**Sleep**
1.4	1.4	–	1.4		
Idle	–	–	1.4	–	

Two versions of our solution were considered. They are respectively referred to as LMDP-raw and LMDP-add. In the first version packet prioritization is entirely applied by the underlying application. Whereas in the second version, an additional sojourn time-based priority level is assigned by LMDP to each packet. If so, Eq. (27) is used as an extension of Eq. (18). This scheme aims to increase fairness and mitigate packet loss. We set $\widehat{w}$ to be 50 ms.

$$\theta_k = \tilde{\theta}_k + \left\lceil \frac{w_k}{\widehat{w}} \times \hat{\theta} \right\rceil \tag{27}$$

As a reminder, $w_k, \widehat{w},$ and $\hat{\theta}$ respectively denote packet k's sojourn time, the waiting time threshold and the maximum priority index value; while $\tilde{\theta}_k$ is the priority index assigned by the underlying application to packet k .

The results were compared to those we obtained with PriTranCon-MAC by Nguyen *et al.* [24].

We evaluated all these protocols via four metrics, Throughput, Packet Delivery Ratio (PDR), fairness, and network lifetime [56].

To do so, we randomly and uniformly deployed sensor nodes around a sink varying their population using, a 50 steps scale as described in Table I.

We investigated how the number of nodes influenced these three metrics.

Jain's fairness index [56] was used to evaluate fairness. In our case, the latter index assesses time slot allocation strategy of each protocol, considering throughput.

This index was calculated during each round and averaged at the end of the experiment (when the first node died).

Link quality was varied using uniform distribution to change randomly the parameters described in Table IV.

M/M/1 queuing model was used for nodes' buffers. Arrival rates were randomly chosen in intervals [50,200] and [30,250] kb/s respectively for sensor nodes and the sink to generate heterogeneous and burst traffic flows. The nodes' buffer size was set to 250 MB.

Any packet was dropped if its sojourn time exceeded 60 ms.

A priority index was assigned to each data packet on arrival in the sensor nodes' buffer. The latter index was randomly and uniformly picked in interval [1, 4] with 25% for each level of priority.

This experiment was replicated 60 times for each variation of the number of nodes. Results were averaged with a 95% confidence interval.

The experiment started after all the nodes were deployed and ended according to the network lifetime definition i.e., when a sensor or all the sensors depleted their energy.

V. RESULTS AND DISCUSSION

In this section, we analyse and explain the results we obtained from experiments described in the previous one.

A. Packet Delivery Ratio

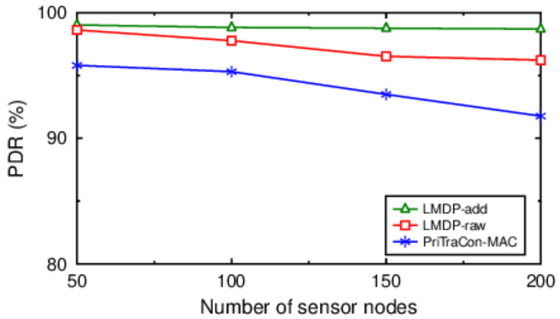

Figure 7. Packet delivery ratio vs. number of sensor nodes.

Fig. 7 shows that the three protocols have ratios higher than 90% irrespective of network size. However, ratios obtained with PriTraCon-MAC tend to decline as the number of sensor nodes grows. This is because PriTraCon-MAC is actually optimized for only small-size one-hop neighborhoods. Indeed, when there are more senders only one RTS beacon is accepted by cycle while other RTSs are delayed. Such a delay often contributes to the dropping of packets. By contrast, LMDP-raw and LMDP-add yield higher values (an improvement of 5.06% and 3.41% on average) due to their ability to select both senders and the amount of data to forward according to the sink's demand (i.e., the remaining space of its buffer). In addition, LMDP-add specifically avoids sojourn time extensions by including this parameter when setting packets' priorities.

B. Throughput

Fig. 8 suggests that the three protocols provide good throughput irrespective of the network size.

PriTraCon-MAC values decrease with the number of sensors. This is also caused by the delay of RTSs in each cycle even for low-priority packets. However, LMDP-raw and LMDP-add increase the throughput respectively by 26.27% and 35.20%; this is also due to their ability to particularly adapt to the sink's service rate. Unlike LMDP-raw, LMDP-add helps sensor nodes send even very small very small bursts provided that their sojourn time reaches the defined threshold. This behavior tends to slightly lower the average throughput irrespective of the network size.

C. Fairness

Fig. 9 shows that LMDP-raw and LMDP-add improve the fairness by 19.35% and 32.49% compared with PriTraCon-MAC. LMDP-add yields higher ratios (95% on average) due to its sojourn time-based packet priority scheme. By contrast, LMDP-raw and PriTraCon-MAC are queue length-oriented and leverage packet priorities defined by the underlying application. To break ties, these protocols tend to consider only bursts size irrespective of their sojourn time. But with PriTraCon-

MAC the delayed RTSs increase packet drop rates and thus prevent some nodes from sending any data. This situation gets worse as the number of sensor nodes grows.

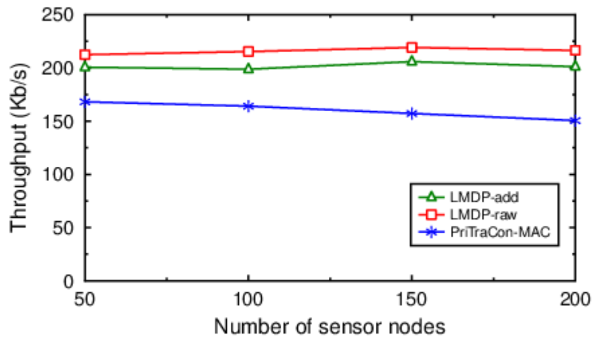

Figure 8. Throughput vs. number of sensor nodes.

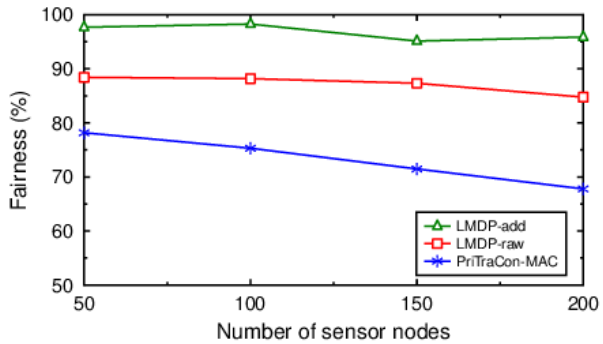

Figure 9. Jain's Fairness Index vs. Number of sensor nodes.

D. Network Lifetime

Fig. 10 shows that LMDP-raw and LMDP-add respectively increase network lifetime by 33.74% and 60% compared with PriTraCon-MAC. Indeed, the higher the number of sender nodes higher is the number of retransmissions due to packet drops. However, these energy-consuming retransmissions are more frequent with PriTraCon-MAC due to high packet loss rates caused by RTSs delays.

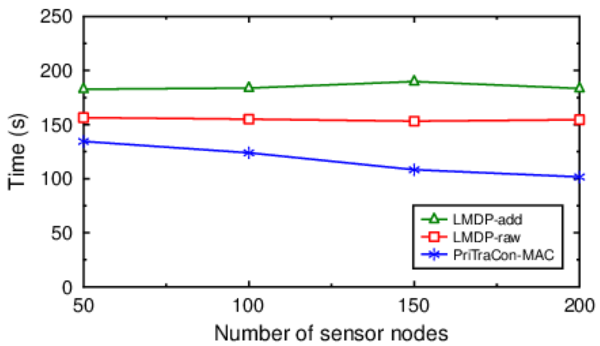

Figure 10. Lifetime (Until first node dies) vs. Number of sensor nodes.

Additionally, unlike the two versions of LMDP, PriTraCon-MAC does not consider link state variations. Therefore, this protocol is more sensitive to packet losses due to poor links and their inherent retransmissions. LMDP-add specifically prevents delays due to sojourn time overrun thus packet retransmissions.

VI. Conclusion

In this paper, we presented LMDP (Last Mile Delivery Protocol), a TDMA-based solution for addressing the hot spot problem i.e., the fast energy drains in the sink's one-hop neighbourhood. This question was formulated as a combination of two well-known optimization problems namely, the precedence-constrained knapsack problem and the single runway aircraft landing one. We proposed two mixed-integer linear programs respectively solved by a genetic algorithm-based heuristic and a specific one. LMDP leverages nodes' congestion degree, link state, and single queue-based packet prioritization to select nodes and schedule data transmissions. In comparison with PriTraCon-MAC (a related IEEE 802.14.5-based solution), LMDP shows better performance in terms of fairness, packet delivery, throughput, and network lifetime. In future work, we will focus on congestion control in the whole network.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization of the idea by G.H.F.D. and A. K. A. Supervision by T. N.; Research, implementation, and analysis during different stages G.H.F.D. and A. K. A. G.H.F.D. and A.K.A. wrote the paper; all authors reviewed the paper and approved the final version.

References

[1] K. Erciyes, "Case study: Environment monitoring by a wireless sensor network," in *Computer Communications and Networks*. Springer International Publishing, 2019, pp. 299–328.

[2] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Applied System Innovation*,vol. 3, no. 1, p. 14, Feb. 2020.

[3] J. Lloret, S. Sendra, L. Garcia, and J. M. Jimenez, "A wireless sensor network deployment for soil moisture monitoring in precision agriculture," *Sensors*, vol. 21, no. 21, p. 7243, Oct. 2021.

[4] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in internet of things (IoT)," in *Proc. 1st Inter. Conf. on Comput. in Materials and Applied Engineering*, Uttarakhand (India), 2021, pp. 161–165.

[5] K. Guravaiah, A. Kavitha, and R. L. Velusamy, "Data collection protocols in wireless sensor networks," *Wireless Sensor Networks - Design, Deployment and Applications*, IntechOpen, 2021, pp 1–20.

[6] R. Zagrouba and A. Kardi, "Comparative study of energy efficient routing techniques in wireless sensor networks," *Information*, vol. 12, no. 1, p. 42, Jan. 2021.

[7] A. Jain and B. Reddy, "Sink as cluster head: An energy efficient clustering method for wireless sensor networks," in Proc. *Inter. Conf. on Data Mining and Intelligent Computing*, Delhi (India), 2014, pp.1–6.

[8] H. U. Yildiz, "Utilization of multi-sink architectures for lifetime maximization in underwater sensor networks," in *Proc. 2nd IEEE Middle East and North Africa COMMunications Conf.*, Manama (Bahrain), 2019, pp. 1–5.

[9] J. Li, Q. Han, and W. Wang, "Characteristics analysis and suppression strategy of energy hole in wireless sensor networks," *Ad Hoc Networks*, vol. 135, p. 102938, Oct. 2022.

[10] M. S. Akbar, Z. Hussain, M. Sheng, and R. Shankaran, "Wireless body area sensor networks: Survey of MAC and routing protocols

for patient monitoring under IEEE 802.15.4 and IEEE 802.15.6," *Sensors*, vol. 22, no. 21, p. 8279, Oct. 2022.

[11] A. N. Sakib, M. Drieberg, S. Sarang, A. A. Aziz, N. T. T. Hang, and G. M. Stojanovi´c, "Energy-aware QoS MAC protocol based on prioritizeddata and multi-hop routing for wireless sensor networks," *Sensors*, vol. 22, no. 7, p. 2598, Mar. 2022.

[12] W. Zhao, Y. Li, B. Yan, and L. Xu, "Median access control protocols for sensor data collection: A review," *IEEE Access*, vol. 8, pp. 160 078–160 098, Aug. 2020.

[13] F. Afroz and R. Braun, "Energy-efficient MAC protocols for wireless sensor networks: a survey," *International Journal of Sensor Networks*, vol. 32, no. 3, p. 150, Feb. 2020.

[14] F. A. Alfouzan, "Energy-efficient collision avoidance MAC protocols for underwater sensor networks: Survey and challenges," *Journal of Marine Science and Engineering*, vol. 9, no. 7, p. 741, Jul. 2021.

[15] A. S. Sadeq, R. Hassan, H. Sallehudin, A. H. M. Aman, and A. H. Ibrahim, "Conceptual framework for future WSN-MAC protocol to achieve energy consumption enhancement," *Sensors*, vol. 22, no. 6, p. 2129, Mar. 2022.

[16] M. Youn, Y.-Y. Oh, J. Lee, and Y. Kim, "IEEE 802.15.4 based QoS support slotted CSMA/CA MAC for wireless sensor networks," in *Proc. Inter. Conf. on Sensor Technologies and Applications*, 2007, Valencia (Spain), pp. 113–117.

[17] L. Alkama and L. Bouallouche-Medjkoune, "IEEE 802.15.4 historical revolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, Oct. 2020.

[18] A. R. Urke, Ø. Kure, and K. Øvsthus, "A survey of 802.15.4 TSCH schedulers for a standardized industrial internet of things," *Sensors*, vol. 22, no. 1, p. 15, Dec. 2021.

[19] N. Choudhury, R. Matam, M. Mukherjee, and J. Lloret, "A performance-to-cost analysis of IEEE 802.15.4 MAC with 802.15.4e MAC modes," *IEEE Access*, vol. 8, pp. 41936–41950, Feb. 2020.

[20] I. U. Onwuegbuzie, S. A. Razak, I. F. Isnin, T. S. J. Darwish, and A. Aldhaqm, "Optimized backoff scheme for prioritized data in wireless sensor networks: A class of service approach," *PLOS ONE*, vol. 15, no. 8, p. e0237154, Aug. 2020.

[21] I. Bouazzi, M. Zaidi, M. Usman, and M. Z. M. Shamim, "A new medium access control mechanism for energy optimization in WSN: traffic control and data priority scheme," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, Feb. 2021.

[22] Q. Huamei, F. Linlin, Y. Zhengyi, Y. Weiwei, and W. Jia, "An energyefficient MAC protocol based on receiver initiation and multi-priority backoff for wireless sensor networks," *IET Communications*, vol. 15, no. 20, pp. 2503–2512, Oct. 2021.

[23] I. A. M. Gonzalez and V. Turau, "A virtual sink-based strategy for reducing the funneling effect in IEEE 802.15.4 DSME networks," in *Proc. 18th International Conference on Distributed Computing in Sensor Systems*, 2022, Los Angeles (USA), pp. 21–24.

[24] T. C. Nguyen, H.-C. Le, S. Sarang, M. Drieberg, and T.-H. T. Nguyen, "Priority and traffic-aware contention-based medium access control scheme for multievent wireless sensor networks," *IEEE Access*, vol. 10, pp. 87 361–87 373, Aug. 2022.

[25] T. Hai, J. Zhou, T. V. Padmavathy, A. Q. Md, D. N. A. Jawawi, and M. Aksoy, "Design and validation of lifetime extension low latency MAC protocol for wireless sensor networks using a hybrid algorithm," *Sustainability*, vol. 14, no. 23, p. 15547, Nov. 2022.

[26] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985–997, May 2009.

[27] F. R. Yazdi, M. Hosseinzadeh, and S. Jabbehdari, "A priority-based MAC protocol for energy consumption and delay guaranteed in wireless body area networks," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1677–1696, May 2019.

[28] S. Borbash and A. Ephremides, "Wireless link scheduling with power control and SINR constraints," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5106–5111, Nov. 2006.

[29] M. Bakshi, B. Jaumard, M. Kaddour, and L. Narayanan, "On TDMA scheduling in wireless sensor networks," in *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering*, 2016, Vancouver (Can.), pp. 1–6.

[30] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[31] S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, vol. 5, pp. 4298–4328, Feb. 2017.

[32] A. Yousaf, F. Ahmad, S. Hamid, and F. Khan, "Performance comparison of various LEACH protocols in wireless sensor networks," in *Proc. IEEE 15th International Colloquium on Signal Processing & Its Applications*, 2019, Penang (Malysia), pp. 108–113.

[33] I. Daanoune, B. Abdennaceur, and A. Ballouk, "A comprehensive survey on LEACH-based clustering routing protocols in wireless sensor networks," *Ad Hoc Networks*, vol. 114, p. 102409, Apr. 2021.

[34] T. Kaur and D. Kumar, "ETPS-MAC: Energy traffic priority scheduling-based QoS-aware MAC protocol for hierarchical WSNs," *International Journal of Electronics*, vol. 106, no. 9, pp. 1344–1359, Apr. 2019.

[35] T. Kaur and D. Kumar, "FPS-MAC: Fuzzy priority scheduling-based MAC protocol for intelligent monitoring systems," *International Journal of Communication Systems*, vol. 33, no. 15, p. e4540, Jul. 2020.

[36] B. A. Muzakkari, M. A. Mohamed, M. F. A. Kadir, and M. Mamat, "Queue and priority-aware adaptive duty cycle scheme for energy efficient wireless sensor networks," *IEEE Access*, vol. 8, pp. 17231–17242, Jan. 2020.

[37] J. Jeon and H. Park, "Data fairness transmission and adaptive duty cycle through machine learning in wireless sensor networks," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 2, p. 349, Jun. 2022.

[38] F. Lin, W. Dai, W. Li, Z. Xu, and L. Yuan, "A framework of priority-aware packet transmission scheduling in cluster-based industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5596–5606, Aug. 2020.

[39] S. Poudel and S. Moh, "Energy-efficient and fast MAC protocol in UAV-aided wireless sensor networks for time-critical applications," *Sensors*, vol. 20, no. 9, p. 2635, May 2020.

[40] D. Espinoza, M. Goycoolea, and E. Moreno, "The precedence constrained knapsack problem: Separating maximally violated inequalities," *Discrete Applied Mathematics*, vol. 194, pp. 65–80, Oct. 2015.

[41] A. Aslan, E. Ursavas, and W. Romeijnders, "A precedence constrained knapsack problem with uncertain item weights for personalized learning systems," *Omega*, vol. 115, p. 102779, Feb. 2023.

[42] M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Springer US, 1972, pp. 85–103.

[43] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*.Springer Berlin Heidelberg, 2003, pp. 147–163.

[44] S. Laabadi, M. Naimi, H. E. Amri, and B. Achchab, "The 0/1 multidimensional knapsack problem and its variants: A survey of practical models and heuristic approaches," *American Journal of Operations Research*, vol. 08, no. 05, pp. 395–439, Sep. 2018.

[45] R. P. Singh, "Solving 0-1 knapsack problem using genetic algorithms," in *Proc. IEEE 3rd Inter. Conf. on Communication Software and Networks*, 2011, Xi'an (China), pp. 591–595.

[46] T. Pradhan, A. Israni, and M. Sharma, "Solving the 0-1 knapsack problem using genetic algorithm and rough set theory," in *Proc. IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, Ramanathapuram (India), pp. 1120–1125.

[47] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft landings-the static case," *Transportation Science*, vol. 34, no. 2, pp. 180–197, May 2000.

[48] 1S. Ikli, C. Mancel, M. Mongeau, X. Olive, and E. Rachelson, "The aircraft runway scheduling problem: A survey," *Computers & Operations Research*, vol. 132, p. 105336, Aug 2021.

[49] R. Prakash, R. Piplani, and J. Desai, "An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 570–581, Oct. 2018.

[50] G. S. Veresnikov, N. A. Egorov, E. L. Kulida, and V. G. Lebedev, "Methods for solving of the aircraft landing problem. i. exact solution methods," *Automation and Remote Control*, vol. 80, no. 7, pp. 1317–1334, Jul. 2019.

[51] G. S. Veresnikov, N. A. Egorov, E. L. Kulida, and V. G. Lebedev, "Methods for solving of the aircraft landing problem. ii. approximate solution methods," *Automation and Remote Control*, vol. 80, no. 8, pp. 1502–1518, Aug. 2019.

[52] A. Varga, "OMNeT++," *Modeling and Tools for Network Simulation*, Springer, Berlin Heidelberg, 2010, pp. 35–59.

[53] C. A. Boano, M. A. Zuniga, T. Voigt, A. Willig, and K. Romer, "The triangle metric: Fast link quality estimation for mobile wireless sensor networks," in *Proc. IEEE 19th International Conference on Computer Communications and Networks*, 2010, Zurich (Switzerland), pp. 1–7.

[54] B. Bougard, F. Catthoor, D. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives," in *Proc. Design, Automation and Test in Europe*, 2005, Munich (Germany), pp. 196–201.

[55] L. D. Nardis and M.-G. D. Benedetto, "Overview of the IEEE 802.15.4/4a standards for low data rate wireless personal data networks," in *Proc. IEEE 4th Workshop on Positioning, Navigation and Communication*, 2007, Hannover (Germany), pp. 285–289.

[56] M. S. Daas, S. Chikhi, and E.-B. Bourennane, "A dynamic multi-sink routing protocol for static and mobile self-organizing wireless networks: A routing protocol for internet of things," *Ad Hoc Networks*, vol. 117, p. 102495, Jun. 2021.