Energy-delay Aware Data Collection in Mobile Sink-based Wireless Sensor Networks with Optimized Visiting Points and Two-level Data Aggregation

Nadjib Benaouda[1], Boubakeur Moussaoui[1,2], and Oussama Senouci[1]

[1] *University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria,*
[2] *Mohammed Boudiaf University, M'sila, Algeria*

https://doi.org/10.26636/jtit.2026.3.2540

Abstract — This study addresses data collection challenges in wireless sensor networks (WSNs) employing mobile sinks (MSs). Although MSs improve flexibility and scalability, achieving balanced optimization of data collection delay and energy consumption remains challenging, as minimizing one objective typically comes at the expense of the other. To overcome this limitation, we propose the optimized grid-based data collection with mobile sink (ODCMS) approach. In ODCMS, the MS follows a predefined trajectory and visits strategically selected visiting points (VPs) to collect aggregated data. The number of VPs is minimized to reduce tour length and thus latency, while a two-level data aggregation scheme shortens transmission distances and lowers forwarding overhead, thereby decreasing energy consumption. Energy balancing is ensured through periodic rotation of aggregation roles to prevent early node depletion. The MS movement among VPs and the base station (BS) is modeled as a connected graph, and its trajectory is constructed as a Hamiltonian cycle using a backtracking algorithm. ODCMS also supports rapid reporting of urgent events to the BS prior to MS arrival. NS-3 simulations compared with EDEDA and EFDC demonstrate that ODCMS achieves shorter tour length and data collection delay, lower energy consumption, and longer network lifetime than other approaches.

Keywords — data aggregation, grid-based structure, mobile sink, path planning, visiting points, WSNs

1. Introduction

Wireless sensor networks (WSNs) comprise distributed autonomous nodes that monitor physical or environmental conditions and transmit collected data to a central base station (BS) or sink for processing. WSNs are fundamental to modern networks, forming the backbone of the Internet of Things (IoT), with applications in precision agriculture, industrial automation, healthcare, environmental monitoring, and smart cities [1]–[4].

Energy efficiency remains a major challenge in WSNs, as sensor nodes are typically battery-powered. In traditional architectures with static sinks, nodes located near the sink must handle excessive data forwarding, creating hotspot areas [5], [6]. This uneven load results in rapid energy depletion of these nodes, which consequently shortens the network lifetime.

Mobile sinks (MSs) have emerged as a promising solution. Depending on the application, they may take the form of unmanned aerial vehicles (UAVs) [7], [8], unmanned ground vehicles (UGVs) [9], autonomous underwater vehicles (AUVs) [10], or devices carried by humans or animals [11].

A MS can reduce energy consumption, balance energy usage, and extend network lifetime by collecting data from multiple locations [3], [5], [6], [11]. In agricultural WSNs, mobile drones replace fixed sinks to collect aggregated data from sensor nodes and reduce long-range transmissions [7], [12], [13].

Similarly, in forest fire monitoring, drone mobility enables efficient network-wide data collection, reducing energy consumption and reliance on multi-hop communication [14].

This work addresses the periodic collection of data by a MS from all WSN nodes. A straightforward approach requires the MS to visit every node, which results in excessively long collection times due to extensive travel. To overcome this, strategically selected locations, called visiting points (VPs), can be used, enabling the MS to gather data efficiently from nearby sensors.

Although this reduces the collection period, identifying the optimal set of VPs and designing an efficient MS route that simultaneously minimizes both delay and energy consumption remains a challenging task [15].

To reduce data collection delay, it is crucial to minimize the number of VPs required for full network coverage and to establish the optimal trajectory for the MS to visit each VP, starting and returning to the BS.

However, too few VPs may force sensor nodes to transmit data over long distances, either via direct long single-hop transmissions, where nodes increase transmission power to reach the MS, or multi-hop forwarding through intermediate nodes. In both cases, energy consumption rises sharply, accelerating network-wide energy depletion.

In response to these challenges, this paper proposes the optimized grid-based data collection with mobile sink (ODCMS) approach. Its primary goals are to minimize data collection delay, reduce energy consumption, and prevent energy imbalance among sensor nodes. ODCMS achieves this by carefully minimizing the number of VPs required for full network coverage, while employing a hierarchical two-level data aggregation scheme that effectively shortens transmission distances and limits forwarding load. Aggregation roles are periodically rotated to evenly balance energy usage across the network, thereby significantly extending its overall lifetime.

ODCMS organizes the sensor field into a virtual grid, partitioning the deployment area based on node communication ranges and region size. The grid dimensions are defined in accordance with design objectives to support the desired selection of VPs. Each grid cell has a designated grid cell head (GCH) that aggregates data from local nodes. At a higher level, one GCH in each quad of four cells is selected as the grid quad head (GQH), which further aggregates the data and prepares it for collection by the MS.

In addition, strategically chosen grid intersection points serve as VPs. The MS begins its tour at the BS and follows a predetermined trajectory visiting each VP exactly once. Its possible movements between VPs and the BS are modeled as a connected graph, and a valid data collection route is determined as a Hamiltonian cycle using a backtracking algorithm. At each VP, the MS gathers data from the four surrounding quads via a two-level aggregation process. This continues until all VPs are visited, after which the MS returns to the BS with the aggregated data.

To further enhance responsiveness, ODCMS also integrates a complementary mechanism to promptly deliver urgent event data directly to the BS via multi-hop communication, thereby avoiding delays associated with waiting for the MS's arrival.

The main contributions of this work are:

1) **Grid-based clustering** – the sensing field is divided into virtual grid cells, enabling structured clustering and efficient VP selection.
2) **Hierarchical aggregation** – each cell elects a GCH, and one GCH per quad becomes the GQH to deliver aggregated quad data to the MS, reducing transmission distances, forwarding load, and balancing energy via periodic role rotation.
3) **VP selection strategy** – minimizes the number of VPs required for full coverage and thus, reducing data collection delay.
4) **Trajectory planning** – models MS movements as a connected graph and computes a Hamiltonian cycle using a backtracking approach.
5) **Event-driven reporting** – supports urgent data delivery directly to the BS without waiting for the MS tour.
6) **Simulation-based evaluation** – ODCMS is implemented in NS-3 and compared with the state-of-the-art EDEDA [15] and EFDC [16] protocols across multiple performance metrics.

This article is structured as follows. Section 2 reviews related work and discusses their limitations. Section 3 presents the network model along with the key assumptions considered. Section 4 introduces ODCMS, while Section 5 provides its detailed description. Section 6 presents the performance evaluation, and finally, Section 7 concludes the article.

2. Related Works

This section reviews recent related studies, emphasizing indirect data collection through VPs rather than direct node-by-node collection methods [17]–[19].

The authors in [20] proposed VGRSS, a virtual grid-based method for selecting rendezvous points (RPs) and sojourn locations to improve energy efficiency and data collection in WSNs. The approach operates in two phases: the sensor field is partitioned into grid cells, then a grid cell coordinator (GCC) is elected from the cell members using a fuzzy inference system that accounts for residual energy and node centrality. Each GCC aggregates and stores data from associated nodes for the MS. To prevent premature node failure, GCCs' energy is monitored each round to determine if replacements are needed. The MS visits sojourn locations at grid intersections, following a trajectory constructed via the travelling salesman problem using Christofides heuristics [21].

In [15], the authors proposed EDEDA, an energy- and delay-efficient data collection approach based on a virtual grid. The sensing area is divided into grid cells, and a grid cell head (GCH) is elected from grid cell members (GCMs) based on proximity to the cell center and residual energy. A subset of cells is selected as visiting cells (VCs), allowing the MS to collect data from the GCHs of nine adjacent cells in one hop. The MS follows a Hamiltonian cycle starting and ending at the BS, while a GCH replacement strategy prevents premature node failure.

In [22], the authors proposed a virtual grid-based MS data collection method in which each cell elects a GCH. The MS follows paths along grid peripheries and diagonals, selecting one cell per path as a rendezvous grid cell (RGC). In each RGC, a rendezvous GCH (RGCH) broadcasts the MS location, enabling GCHs to adjust minimum-hop routes and efficiently forward aggregated data to the MS.

In [16], the authors proposed EFDC, an energy-efficient data-gathering protocol for WSNs with a MS. The approach employs hierarchical agglomerative clustering (HAC) to form clusters and select the optimal clustering level. CHs are chosen as the nodes closest to cluster centroids. The VPs for the MS are designated as the coverage intersection points (CIPs) located along the boundaries of the cluster communication ranges. The MS travels to these CIPs, halts for a specified duration to collect data, and then continues its trajectory using the traveling salesman problem with neighborhoods (TSPN) algorithm.

In [23], the authors proposed E2OSLMS, a method to determine optimal MS locations within clusters, ensuring balanced energy consumption among nodes and preventing energy holes. The approach assumes a circular sensing area, which is divided into coronas, each containing multiple clusters. Cluster heads are elected based on their distance to the cluster center and residual energy. Four MS traverse a spiral trajectory through the designated clusters and return to the starting point. Within each cluster, the precise MS location is determined from member node positions using a fuzzy inference system, enabling efficient, balanced, and reliable data collection across the network.

In [24], an energy-aware path construction (EAPC) approach is proposed to establish an energy-efficient data-gathering path while enhancing the selection of data collection points (CPs). This construction is led under the constraint of a predetermined path length provided as input. The algorithm comprises three primary phases: initialization, selection of CPs, and path construction. A minimal spanning tree (MST) is established during the initialization phase to minimize the total path length. The collection point selection phase determines CPs using a benefit index that optimizes energy usage while minimizing the forwarding burden on sensor nodes. The most direct route between the BS and the CPs is finally generated during the path construction phase.

Article [25], introduced BIIE, an energy-efficient data collection algorithm for WSNs. The algorithm operates in three phases: initialization, path establishment, and local re-clustering. The network is first divided into clusters, with the maximum number determined using a dichotomy method under MS path length constraints. During path establishment, particle swarm optimization (PSO) identifies optimal Rendezvous Nodes (RNs) and the MS trajectory, minimizing energy consumption imbalance among nodes. Finally, local re-clustering dynamically adjusts cluster structures based on residual energy, dividing or merging clusters to ensure balanced energy usage, thereby enhancing network lifetime and reliability.

A hybrid data collection method combining rendezvous- and cluster-based approaches is presented in [26] to achieve energy efficiency while meeting delivery deadlines for critical packets. Rendezvous-based collection reduces energy by storing data at rendezvous nodes (RNs) until the MS arrives, whereas cluster-based collection forwards data via multi-hop paths, consuming more energy but reducing delay. The method operates in three phases: clustering nodes based on density and residual energy, selecting strategic points near cluster centers, and determining Steiner points that connect these points. PSO groups Steiner points, and MSs visit each to collect data, which is then transmitted to a central collection point.

An optimal RPs selection with MS trajectory construction algorithm is proposed in [27]. This approach, called ORPSTC, begins by constructing a minimum-cost spanning tree (MST) of the WSN graph. The MST is subsequently partitioned into several clusters, and a CH is selected within each cluster to serve as the RP. Upon identifying the RPs, a computational geometry approach is employed to determine the route of the MS linking these RPs. A RPs reselection process is devised to maintain equitable energy consumption among sensor nodes, with the consideration of virtual RPs (VRPs). VRPs are sensor nodes that can directly communicate with the MS.

In [28], an enhanced MS path planning method for homogeneous and heterogeneous WSNs is proposed, supporting single and multiple MSs. The sensing area is divided into equal regions, and nodes are clustered using the proposed stable election algorithm (SEA) to reduce frequent CH rotations.

In homogeneous networks, CHs are selected based on neighbor count, while in heterogeneous networks, residual energy prioritizes super-nodes. CH positions are reported to the BS to compute MS paths. Visiting points are determined via a minimum weighted vertex cover formulation, and MS trajectories are optimized using multi-objective evolutionary algorithms, including ACO, GA, and SA.

The authors of [29] outline the following method named SCTR to ascertain a minimum number of RPs. The WSN was represented as an undirected graph G. They subsequently determined the connected components of G through the application of breadth-first search (BFS) traversal.

The reasoning for this is to diminish the problem's size, hence decreasing the computing complexity of the solution. The larger connected components are then partitioned into manageable units (MUs) using the combined use of a heuristic and Tarjan's technique. Upon acquisition of these MUs, spectral clustering is employed to create clusters from the MUs. The centroid of each cluster is then identified as RP. The identified RPs are ultimately employed to construct the traveling salesman tour (TSP) for the MS utilizing Christofides' algorithm.

2.1. Discussion

Focusing on minimizing energy consumption and data collection latency, the reviewed articles can be grouped into three main classes, as outlined in Tab. 1.

- **Low/medium energy – medium/high latency:** Methods such as EDEDA, VGRSS, E2OSLMS, SCTR, EFDC, and [28] conserve energy by reducing multi-hop transmissions and control overhead but may increase latency due to more VPs and node waiting time. Energy consumption may also depend on cluster size, as larger clusters increase transmission distances between members and their CHs.
- **High energy – low latency:** Approaches such as TARA and ORPSTC lower data delivery delay by minimizing VPs or maintaining continuous shortest multi-hop paths, at the cost of increased energy from frequent forwarding and control messages.
- **Adaptive/variable solutions:** Methods like EAPC, BIIE, and HRCM balance energy and latency according to application requirements.

Despite existing proposals, jointly optimizing data collection delay and energy consumption remains challenging, as improving one often increases the other. Proposed approach tackles this trade-off by reducing the number of VPs to shorten delays and limiting long-distance and multi-hop transmissions to save energy, effectively balancing both objectives.

Tab. 1. Summary of the reviewed articles.

Method	Structure type	Multi-sink support	Path selection strategy	Energy consumption	Data acquisition latency
EDEDA [15]	Grid	No	Hamiltonian path planning strategy	Low	Medium
EFDC [16]	Cluster	No	Traveling salesman problem with neighborhoods (TSPN)	Low	Medium
VGRSS [20]	Grid	No	Christofides algorithm	Low	Medium
TARA [22]	Hybrid (grid & tree)	No	Computation-free path strategy	High	Low
E2OSLMS [23]	Ring	Yes	Computation-free path strategy	Low	Medium
EAPC [24]	Cluster	No	Convex polygon algorithm	Variable	Variable
BIIE [25]	Cluster	No	Particle swarm optimization	Variable	Variable
HRCM [26]	Cluster	Yes	Dijkstra shortest path algorithm	Variable	Variable
ORPSTC [27]	Cluster	No	Author-defined computational geometry approach	High	Low
Approach in [28]	Cluster	Yes	Multi-objective evolutionary algorithm (MOEA)	Low	Medium
SCTR [29]	Cluster	No	Christofides algorithm	Low	Medium

3. Network Model and Assumptions

This study adopts the following network model and assumptions:

- n stationary sensor nodes are uniformly and randomly deployed in a square area of side L.
- Each node has a maximum communication range R_c, forming a unit-disk graph (UDG), where two nodes i and j are directly connected if their Euclidean distance
$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$
satisfies $d(i,j) \leqslant R_c$.
- A single MS collects data from sensor nodes by visiting predefined VPs, starting and ending at a fixed BS, which stores the collected data.
- Each node knows its location and starts with equal residual energy.
- The MS can reliably communicate with nodes within range and has sufficient energy to complete its tour.
- The MS trajectory is unobstructed.

4. Overview of the Proposed Approach

ODCMS is guided by the following design principles. The network is partitioned using a $k \times k$ virtual grid, based on area size L and node range R_c. The value of k is required to be a multiple of four in order to support the desired selection of VPs. VPs are chosen at grid intersections, each covering sixteen adjacent cells (four quads), as shown in Fig. 1.

Each cell elects a GCH for local aggregation. Among the four GCHs in a quad, the one closest to the VP becomes the GQH, which aggregates data from the other GCHs and sends it to the MS. GCHs and GQHs are periodically re-elected to prevent early failures.

The MS tours the network from the BS, visiting each VP to collect aggregated data, then returns to the BS. The trajectory is modeled as a connected graph, and a Hamiltonian cycle is computed via backtracking. Urgent data are sent directly to the BS without waiting for the MS.

5. In-depth Description of the Proposed Approach

The proposed approach operates through seven key operations. The first step consists of designing the virtual grid. The second identifies the set of VPs, while the third determines how GCHs and GQHs are selected. The fourth operation

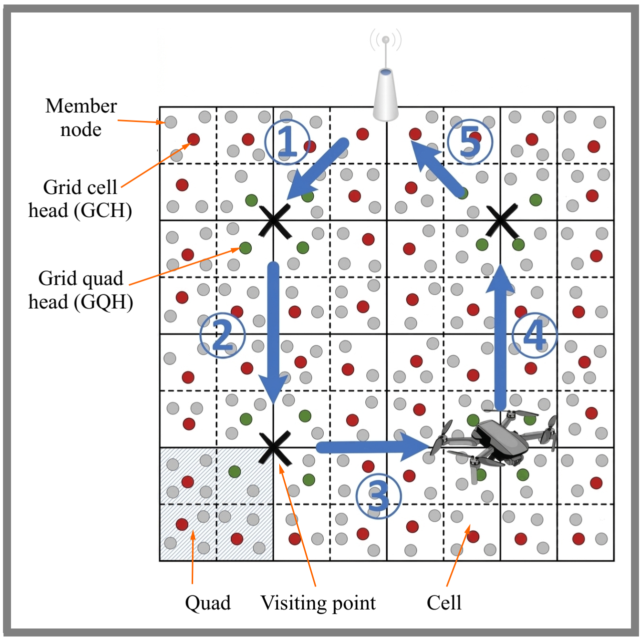

Fig. 1. Overview of ODCMS.

Tab. 2. List of messages used.

Message	Description
GCH_ELECTION	Initiates the election of GCHs/GQHs
GCH_ANNOUNCEMENT	Used by GCHs/GQHs to announce their roles
ASKING_DATA	Used by the MS to trigger data collection from nodes near a VP
DATA	Used to transmit data through the hierarchy from nodes to the MS
GCH_REELECTION	Used by GCHs/GQHs when they are unable to fulfill their designated roles
URGENT_EVENT	Used by sensor nodes to inform the BS of an emergency event

defines the trajectory of the MS, and the fifth explains the data-gathering process. The sixth describes the reelection mechanism for GCHs and GQHs. The seventh addresses the strategy for handling urgent events. These operations are elaborated further in the following sections.

The messages supporting the implementation of proposed solution, are listed in Tab. 2.

5.1. Virtual Grid Design

The virtual grid is created by partitioning the sensing area into square cells based on the nodes' communication range and the sensing area's dimensions.

The grid consists of k rows and k columns, forming a $k \times k$ square grid. The value of k is determined using the adjustment factor α:

$$\alpha = \frac{2\sqrt{2}L}{R_c} \,. \tag{1}$$

Given the value of α, the number k can be identified. It must be a multiple of four to allow the selection of the VPs as desired. It is defined as follows:

$$k = \begin{cases} 4, & \text{if } \alpha \leqslant 4 \\ 8, & \text{if } 4 < \alpha \leqslant 8 \\ 12, & \text{if } 8 < \alpha \leqslant 12 \\ 16, & \text{if } 12 < \alpha \leqslant 16 \\ \dots \end{cases} . \tag{2}$$

It is worth emphasizing that this grid formation ensures communication among nodes in neighboring cells.

To make possible the execution of the subsequent steps, the cell identifier $IdCell_i$ of every sensor node i needs to be calculated. This calculation is made using the coordinates (x_i, y_i) of i, according to the following formula:

$$IdCell_i = pk + q \,, \tag{3}$$

where $p = \lfloor \frac{y_i}{a} \rfloor$ and $q = \lfloor \frac{x_i}{a} \rfloor$, with a corresponding to the side length of a cell.

The value of a is calculated using the following formula:

$$a = \frac{L}{K} \,.$$

Figure 2 depicts the visual representation of an 8×8 virtual grid, showcasing the distinct identifiers assigned to each individual cell.

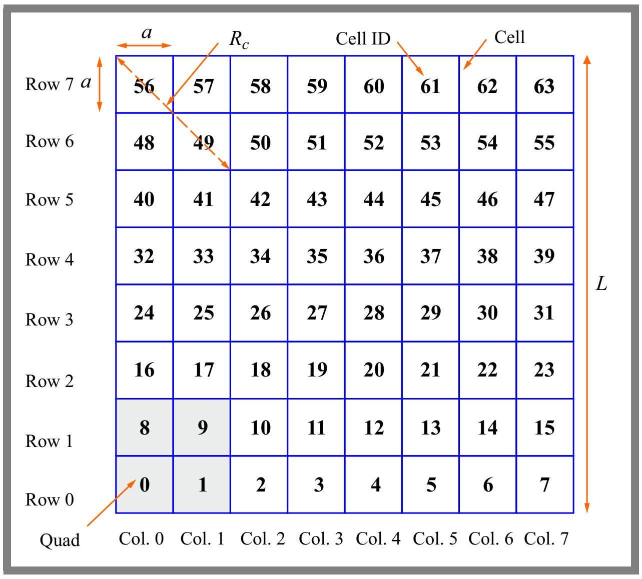

Fig. 2. Virtual grid design.

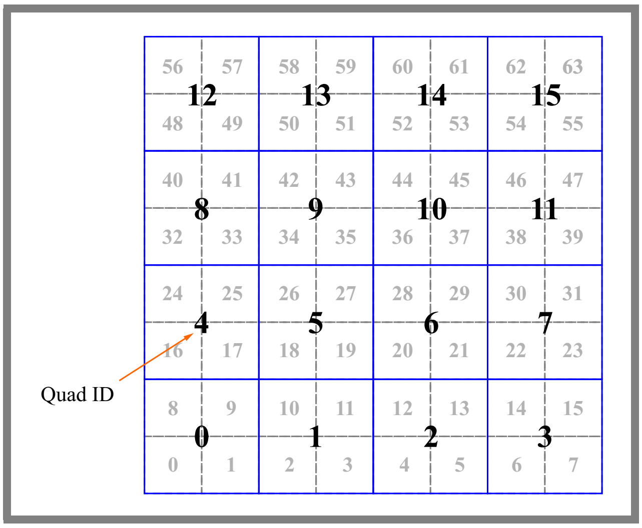

Fig. 3. Identification of the grid's quads.

In addition to cell identifiers, each sensor node i must know the identifier of the quad $IdQuad_i$ to which it belongs. Based on its coordinates, node i computes it as follows:

$$IdQuad_i = p'\frac{k}{2} + q' \,, \tag{4}$$

where $p' = \lfloor \frac{y_i}{2a} \rfloor$ and $q' = \lfloor \frac{x_i}{2a} \rfloor$.

Figure 3 illustrates the identifier assigned to each quad in an 8×8 grid.

5.2. Visiting Points Identification

Several specific intersection points between four neighboring cells are selected as VPs. Each VP is chosen to enable the MS to cover 16 adjacent cells, corresponding to four adjacent quads. The identification of these VPs is performed using Algorithm 1, which is based on the following formula:

$$m = 2a(2z + 1) \,, \tag{5}$$

where $z = 0, 1, \dots, k/4 - 1$.

Algorithm 1 Selection of VPs

Input: a, k
Output: φ: list of identified VPs coordinates

$\varphi \leftarrow \emptyset$
$i \leftarrow 0$
while $i < \frac{k}{4}$ **do**
  $x \leftarrow 2 \times a \times (2 \times i + 1)$
  $j \leftarrow 0$
  while $j < \frac{k}{4}$ **do**
    $y \leftarrow 2 \times a \times (2 \times j + 1)$
    $\varphi \leftarrow \varphi \cup \{(x, y)\}$
    $j \leftarrow j + 1$
  end while
  $i \leftarrow i + 1$
end while
Return φ

The possible values of m correspond to the potential X-intercept and Y-intercept values of the distinct coordinates of the determined VPs, which are henceforth denoted by the set φ. Figure 4a-b illustrates the coordinates of all VPs identified in 8×8 and 12×12 grids, respectively, by applying the Algorithm 1. For example, in a 8×8 grid ($k = 8$), m can take two values: $2a$ and $6a$ according, respectively, to the following values of z (i or j): 0 and 1. As a result, the different coordinates of the four identified VPs are as follows: $(2a, 2a)$, $(6a, 2a)$, $(2a, 6a)$ and $(6a, 6a)$.

5.3. GCHs and GQHs Election

Once the virtual grid is established, clusters naturally form, enabling the selection of GCHs and GQHs. Upon network deployment, each node broadcasts a `GCH_ELECTION` message containing its ID, coordinates, cell, and quad identifiers. Neighboring nodes record this information to maintain awareness of nearby nodes, including those within the same cell.

Within each cell, the node closest to the geometric center is elected as the GCH, responsible for local data collection and initial aggregation. In each quad, the GCH of the cell nearest the designated VP is elected as the GQH, serving as a rendezvous node that further aggregates quad data and delivers it to the MS.

This central positioning reduces transmission power requirements for cluster members and ensures GQHs remain near their associated VPs. Elected GCHs/GQHs broadcast a `GCH_ANNOUNCEMENT` to notify neighbors of their roles.

While GCHs can transmit directly to the MS, forwarding data through GQHs reduces transmission distances, aggregates correlated data, conserves energy, and lowers the relaying workload, thereby improving network efficiency.

5.4. Path Planning

The MS starts and ends its tour at the BS, visiting all VPs along a pre-established path. To determine this route, the BS models possible MS movements between VPs as a connected graph $G' = \{V', E'\}$, where V' includes all VPs and the BS, and E' represents feasible direct movements, as illustrated in

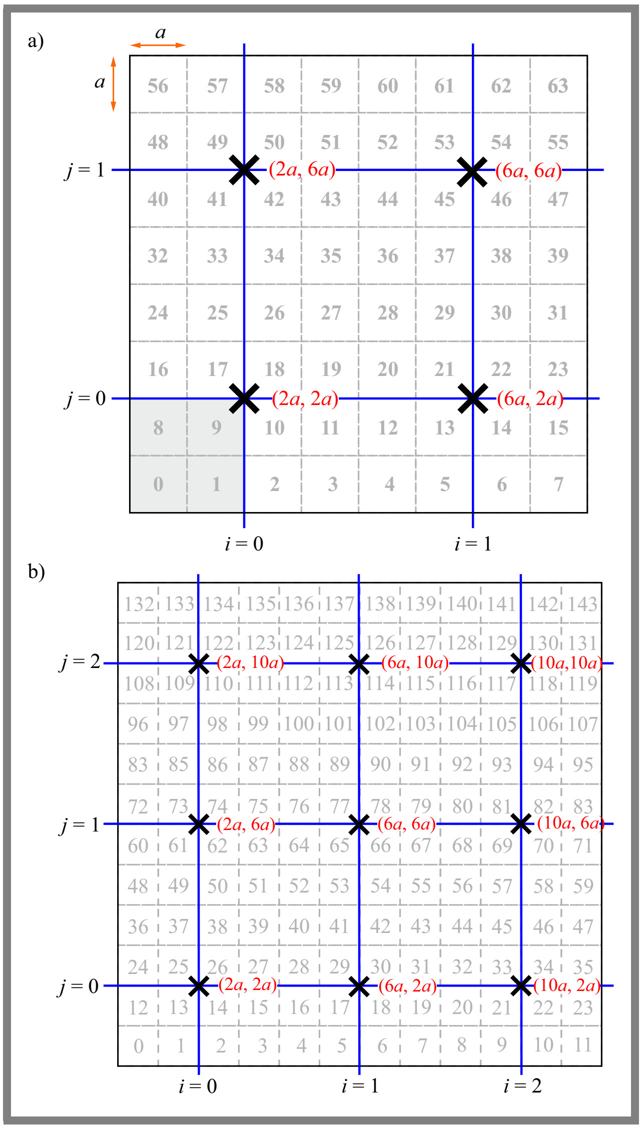

Fig. 4. VPs identification for various grid sizes: a) 8×8 grid and b) 12×12 grid.

Fig. 5, which shows the connected graph used to determine the MS's path in a 12×12 grid.

A valid Hamiltonian cycle, visiting each VP exactly once, is then identified using a backtracking algorithm [30], though other optimization methods can also be applied [3]. The algorithm incrementally builds the cycle by adding adjacent, unvisited vertices and backtracks when no valid extension exists, continuing until a complete cycle is found.

Although finding a Hamiltonian cycle is an NP-complete problem with factorial worst-case complexity, the computational overhead at the BS remains manageable in ODCMS. This is primarily because ODCMS reduces the number of VPs and restricts feasible movements to directly connected neighboring vertices in G', thereby limiting the search space explored by the backtracking algorithm. Consequently, minimizing the number of VPs not only reduces the MS tour length but also alleviates the computational burden of route construction.

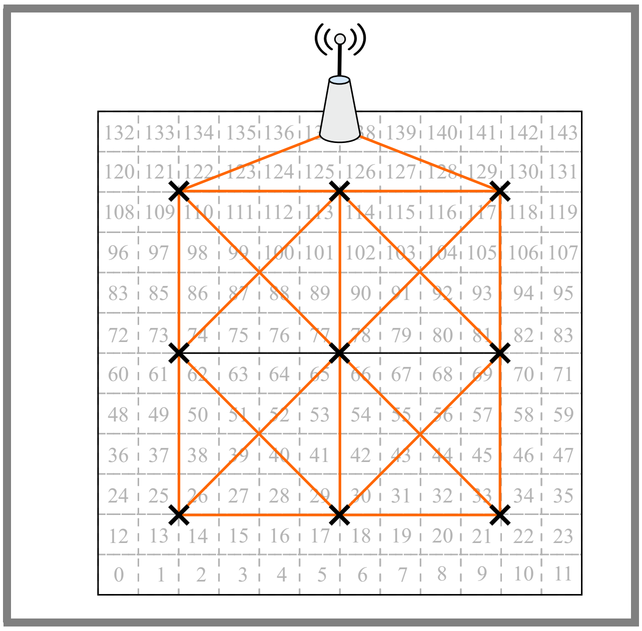

Fig. 5. Connected graph for determining the MS path in a 12 × 12 grid.

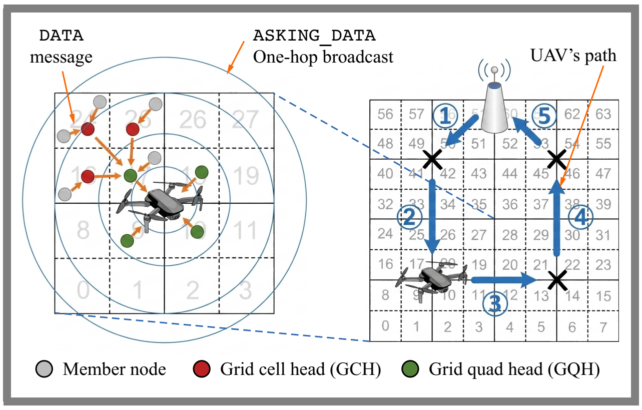

Fig. 6. Data collection process.

5.5. Data Collection

The MS periodically follows the predetermined Hamiltonian cycle, broadcasting an `ASKING_DATA` message at each VP to reach all nodes in the four associated quads. GCHs schedule `DATA` transmissions to their GQH after a short wait to gather member data, and GQHs subsequently forward aggregated data to the MS after a longer wait, ensuring complete collection.

Each `ASKING_DATA` message specifies the targeted cell IDs to limit requests to relevant nodes. At each VP, the MS collects data from the four neighboring GQHs, then moves to the next VP, repeating the process until returning to the BS.

Figure 6 illustrates this data collection procedure.

5.6. Re-election of GCHs and GQHs

GCHs and GQHs are periodically reelected to prevent premature failure and ensure continuous data collection. When the residual energy of a GCH or GQH drops below a predefined threshold $E_{\text{threshold}}$, the node initiates a reelection process by broadcasting a `GCH_REELECTION` message. The current GCH/GQH is excluded from candidacy, and a new leader is selected from among the nodes closest to the cell center that possess sufficient residual energy.

Tab. 3. Simulation parameters used to analyze the proposed system.

Parameter	Value
Sensing field dimensions $L \times L$	120 ×v120, 160 × 160, 200 × 200, 240 × 240, 280 × 280, 320 × 320 m^2
Number of sensor nodes	200 – 400
Location of BS	$(\frac{L}{2}, 0)$
Communication range R_c	50 m
Initial energy per node	1 J
Energy threshold $E_{\text{threshold}}$	40%
Data packet size	512 bytes
Mobile sink speed	10 m/s
Duration between sink tours T_{rest}	1 hour
Sink sojourn duration in a VP	10 s
Simulation duration	Until first node dies
E_{elec}	50 nJ/bit
E_{mp}	0.0013 pJ/bit/m^4
E_{Fs}	10 pJ/bit/m^2

5.7. Urgent Events Handling

In emergency scenarios, when a sensor node detects a measurement exceeding a predefined threshold, it encapsulates the measurement within an emergency notification message (`URGENT_EVENT`) and transmits it to the GCH of its respective cell. To mitigate false alarms arising from sensor faults or anomalies, the GCH validates the urgency of the event by requiring the reception of multiple corroborative `URGENT_EVENT` messages from distinct member nodes within the same cluster.

Upon validation, the GCH initiates a unicast transmission of the `URGENT_EVENT` message directly to the BS employing a geographic routing protocol, thereby bypassing the inherent latency associated with awaiting the MS arrival. This direct transmission strategy significantly reduces the end-to-end latency for critical event reporting within the network.

6. Performance Evaluation

6.1. Simulation Setup

To evaluate the proposed approach, ODCMS was implemented in NS-3, incorporating its message types, data structures, and protocol operations. For comparison, EDEDA [15] was also implemented, as it similarly employs a virtual grid and mobile sink, with the sink visiting selected visiting cells (VCs) to collect data from nearby nodes.

In addition, EFDC [16] was implemented, which employs the HAC algorithm for cluster formation and directs the MS through coverage intersection points (CIPs), serving as VPs for data collection.

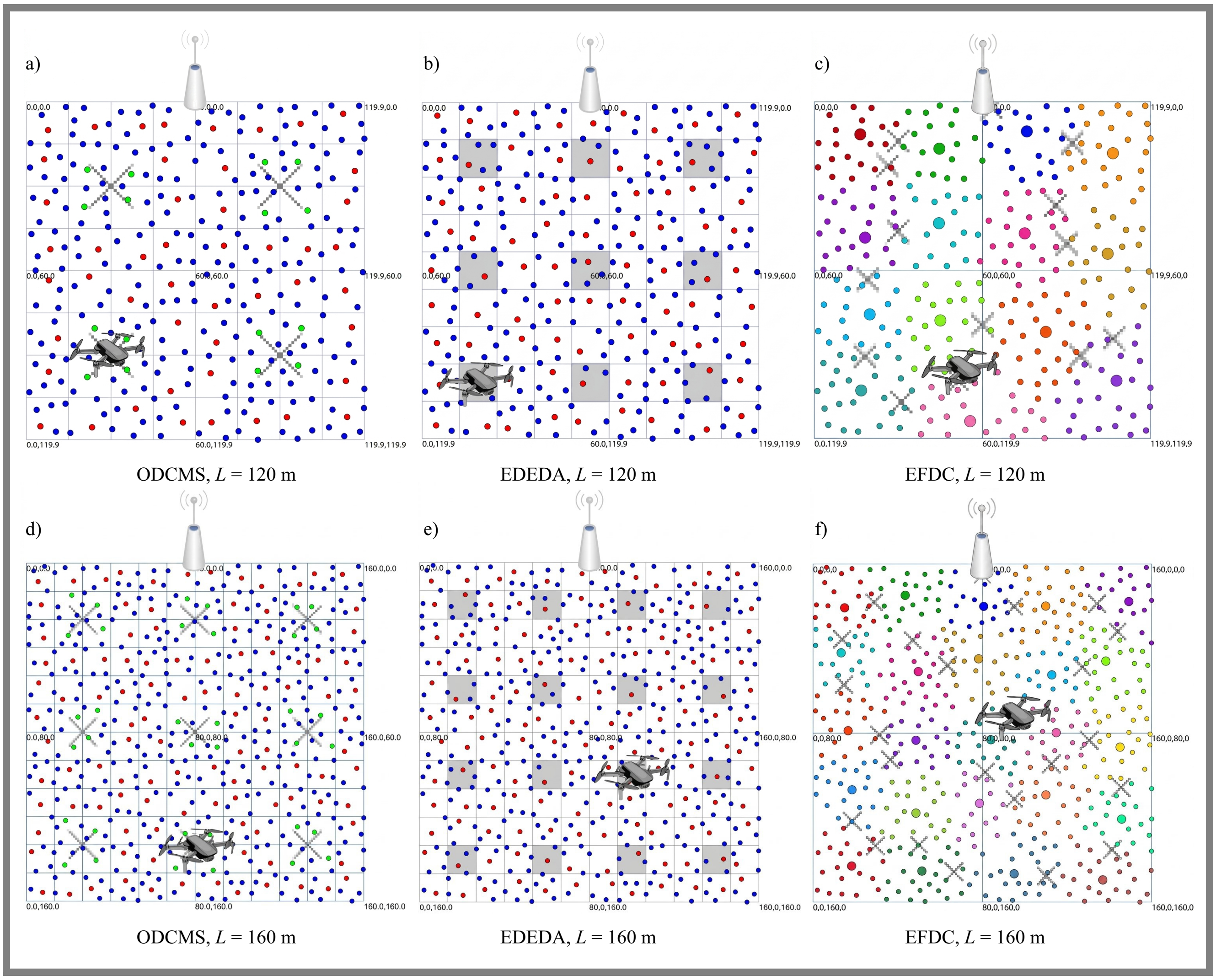

Fig. 7. NetAnim snapshots showing the configuration of the simulated network after the execution of ODCMS, EDEDA, and EFDC for different terrain side lengths.

Extensive simulations were conducted using the parameters in Tab. 3. The number of sensor nodes (200 – 400) was varied to evaluate energy consumption and network lifetime, while the terrain size L was adjusted to assess its impact on MS data collection delay. All other parameters were kept constant.

To reflect a realistic scenario, we considered a fire detection WSN assisted by a drone acting as the MS. Sensors were uniformly deployed, and the drone periodically collected data from designated VPs/VCs before returning to the BS. Data collection occurred hourly, and simulations ran until the first node died.

Before presenting the results, we briefly describe the adopted energy model and the evaluation metrics used for comparison.

6.2. Energy Consumption Model

To evaluate energy consumption, we adopted the same radio energy model as in [20]. The transmission energy for sending a p-bit message over a distance d, denoted by $E_{TX}(p,d)$, and the reception energy $E_{RX}(p)$ are computed as follows:

$$E_{TX}(p,d) = \begin{cases} pE_{\text{elec}} + pE_{\text{fs}}d^2, & d < d_0 \\ pE_{\text{elec}} + pE_{\text{mp}}d^4, & d \geqslant d_0 \end{cases}, \tag{6}$$

$$E_{RX}(p) = pE_{\text{elec}} , \tag{7}$$

where E_{elec} represents the electronics energy per bit, while E_{fs} and E_{mp} denote the free-space and multi-path amplification factors, respectively.

The threshold distance is defined as:

$$d_0 = \sqrt{\frac{E_{\text{fs}}}{E_{\text{mp}}}} . \tag{8}$$

Energy consumed in sensing and processing is neglected, since transmission dominates overall energy dissipation.

6.3. Comparison Metrics

The performance is evaluated using three primary metrics: total energy consumption, network lifetime, and data collection duration. We also report complementary metrics, including energy consumption per message type (`ASKING_DATA` and

DATA), MS tour length, and the number of VPs/VCs, to better explain variations in the main results and provide deeper insight into protocol behavior.

The total energy consumption represents the total energy consumed by all sensor nodes during the simulation. It is computed as the sum of the difference between the initial and residual energy of each node i:

$$E_{\text{total}} = \sum_{i=1}^{n} \left(E_{init_i} - E_{resid_i} \right) . \tag{9}$$

Network lifetime is defined as the period during which the network remains operational before performance degradation occurs, such as node death, disconnection, or coverage loss [31]. In this study, we adopt the commonly used definition: the time from the start of the simulation until the first sensor node depletes its energy.

Data collection duration is defined as the total time required for the MS to depart from the BS, visit all VPs to gather data, and return to the BS. It includes both travel time between VPs and the pause time spent collecting data at each stop. This duration can be expressed as follows:

$$D_{\text{collection}} = \sum_{i=1}^{|\varphi \cup \{\text{BS}\}|} \left(\frac{d(\text{SL}(i), \text{SL}(i+1))}{sink\ speed} \right) + \sum_{j=1}^{|\varphi|} Pt_j \ . \tag{10}$$

The stopping location i is represented by $\text{SL}(i)$, while the pause time or sink stay duration at VP j is denoted by Pt_j.

6.4. Results and Discussion

In the following, we present the simulation results using graphs and NetAnim visualizations, highlighting the performance of ODCMS, EDEDA and EFDC under varying network conditions. Each result represents the average of 10 runs with different random seeds to ensure reliability (Fig. 7).

To ensure fair comparison across scenarios with different durations, total energy consumption is normalized by dividing it by the simulation time (in seconds).

Total energy consumption increases with network density for all protocols, as more nodes generate greater communication traffic and overhead (Fig. 8a). ODCMS consistently consumes less energy than EDEDA. For 200 nodes, ODCMS consumes 9.30 $\times 10^{-5}$ J/s compared with 1.09 $\times 10^{-4}$ J/s for EDEDA, representing a reduction of approximately 14.68%. At 400 nodes, the corresponding values are 2.41 $\times 10^{-4}$ J/s and 2.83 $\times 10^{-4}$ J/s, yielding a reduction of approximately 14.84%.

EFDC exhibits the highest energy consumption, averaging 1.60 $\times 10^{-4}$ J/s at 200 nodes and 4.19 $\times 10^{-4}$ J/s at 400 nodes. This is primarily attributed to its larger clusters, which require member nodes to transmit data over longer distances. The greater number of visiting points in EFDC e.g., 14 at $L =$ 120 m and 24 at $L = 160$ m, further increases overhearing of asking for data messages (Fig. 8b).

The impact of network density on the lifetime of ODCMS, EDEDA, and EFDC under identical conditions, with 200–400 nodes in a fixed 120 $\times$ 120 m area is presented in Fig. 8d.

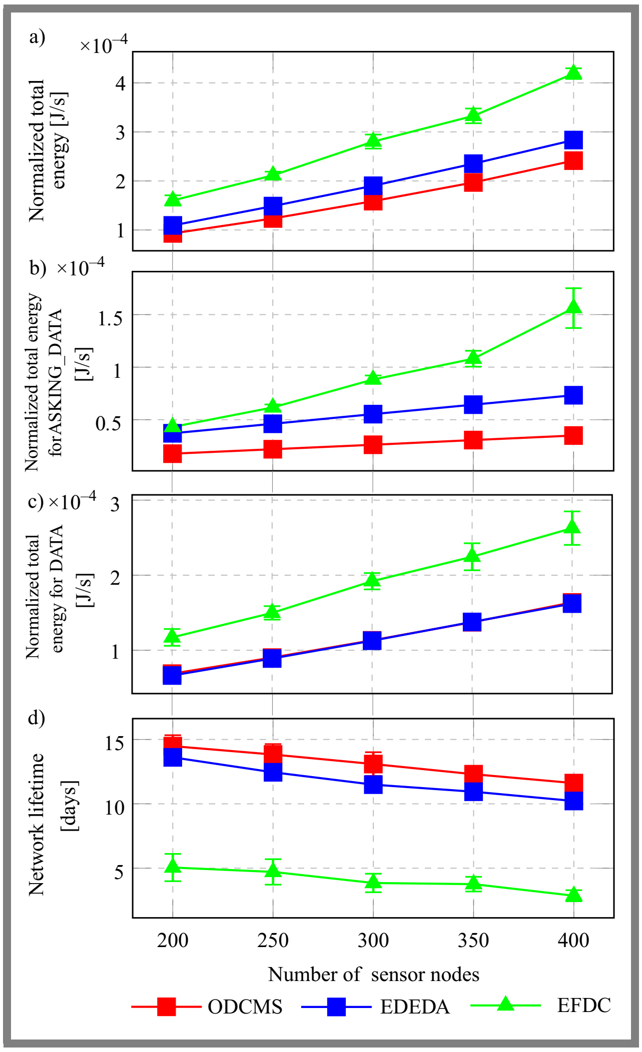

Fig. 8. Comparison of simulation results for ODCMS, EDEDA, and EFDC protocols across multiple scenarios (with standard deviation error bars).

Network lifetime decreases with node density due to increased communication activity and energy depletion. ODCMS consistently outperforms both protocols. At 200 nodes, it achieves 14.48 days, compared with 13.61 days for EDEDA ($\approx$ 6.4% longer) and 5.05 days for EFDC ($\approx 186.73\%$ longer). At 400 nodes, ODCMS reaches 11.61 days, compared with 10.22 days for EDEDA ($\approx$ 13.6% longer) and 2.86 days for EFDC ($\approx 305.94\%$ longer). This advantage stems from ODCMS's improved energy efficiency, while EFDC lacks CH rotation and incurs additional overhead from overhearing data and MS data-request messages.

Figure 9 illustrates the resulting impact on data collection duration for different values of L, while adjusting the number of sensor nodes to maintain a constant density of 2 nodes per 100 m^2. As L varies, the grid size ($k \times k$) changes accordingly, which in turn affects the number of VPs in ODCMS and VCs in EDEDA.

EDEDA and EFDC consistently require longer collection times than ODCMS, with EFDC exhibiting the highest delays. This advantage results from ODCMS minimizing the number of VPs and, consequently, the MS tour length (Fig. 9b). Table 4

Tab. 4. MS tour length, data collection delay, and parameters for ODCMS, EDEDA, and EFDC protocols, along with percentage reductions.

L [m]		120	160	200	240	280	320
Nodes		288	512	800	1152	1568	2048
ODCMS	$k \times k$	8 × 8	12 × 12	12 × 12	16 × 16	16 × 16	20 × 20
	VPs	4	9	9	16	16	25
	Tour [m]	264.85	590.10	737.63	1139.44	1329.35	1958.94
	Delay [min]	1.10	2.48	2.72	4.56	4.83	7.43
EDEDA	$k \times k$	9 × 9	12 × 12	12 × 12	15 × 15	18 × 18	21 × 21
	VCs	9	16	16	25	36	49
	Tour [m]	442.58	759.63	949.54	1469.20	1948.61	2661.71
	Delay [min]	2.24	3.93	4.25	6.62	9.25	12.6
EFDC	VPs	14	24	40	54	68	102
	Tour [m]	336.77	622.36	1023.68	1416.84	1928.43	2624.42
	Delay [min]	2.89	5.04	8.37	11.36	14.55	21.37
Red. (vs. EDEDA)	Tour [%]	40.16	22.32	22.32	22.44	31.78	26.40
	Delay [%]	50.89	36.90	36.00	31.12	47.78	41.03
Red. (vs. EFDC)	Tour [%]	21.36	5.18	27.94	19.58	31.07	25.36
	Delay [%]	61.94	50.79	67.50	59.86	66.80	65.23

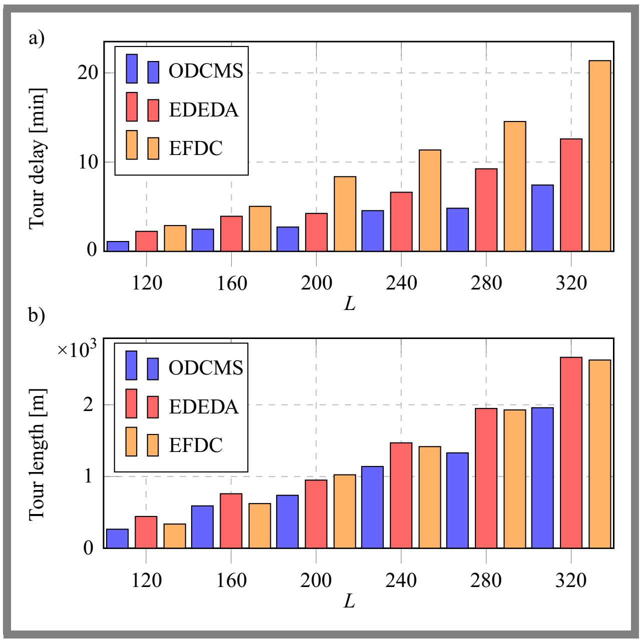

Fig. 9. Tour delay a) and tour length comparisons b) for ODCMS, EDEDA, and EFDC under different terrain dimensions.

confirms that ODCMS selects fewer VPs than both protocols for the same L, resulting in shorter tours and collection times. Although EFDC's tour lengths can be close to those of EDEDA at larger L values, its collection time remains substantially higher because it uses more VPs (14 – 102), leading to more MS stops.

In urgent events handling experiment, we validate ODCMS's urgent event handling by randomly selecting 20 sensor nodes, each generating a single urgent event during a 24-hour simulation, considering two protocol variants. The first variant, ODCMS-direct, sends urgent events immediately to the BS via multi-hop, while ODCMS-deferred stores events at GCHs and forwards them only when the MS collects data. System performance is measured by the end-to-end delay from event generation to BS reception. In ODCMS-direct, this reflects multi-hop forwarding via the GCH, while in ODCMS-deferred, it is dominated by the MS travel time to the VP and back to the BS. In ODCMS-deferred, the MS resting period T_{rest} strongly affects urgent event delay.

Figure 10a shows the CDF of latency for different T_{rest} values, indicating that longer rests lead to higher end-to-end delays, as events must wait for the MS to resume its tour. Figure 10b shows the CDF of latency for ODCMS-direct and ODCMS-deferred ($T_{rest} = 0$). ODCMS-direct exhibits millisecond delays, while ODCMS-deferred reaches minutes due to the MS traveling to VPs, retrieving data, and returning to the BS. These results highlight the necessity of dedicated mechanisms to handle time-critical events when MSs are employed for data collection.

7. Conclusions

This study addressed fixed-path data collection in WSNs using a MS. We proposed ODCMS, which reduces collection delay and energy consumption while extending network lifetime by minimizing VPs and mitigating costly multi-hop and long-distance transmissions via a two-level data aggregation scheme. ODCMS combines virtual grid-based clustering, selective VP selection, hierarchical aggregation, and backtracking-based MS path planning, with urgent events sent directly to the BS.

NS-3 simulations comparing ODCMS with EDEDA and EFDC show improved tour length, delay, energy efficiency, and lifetime. Future work will investigate more realistic

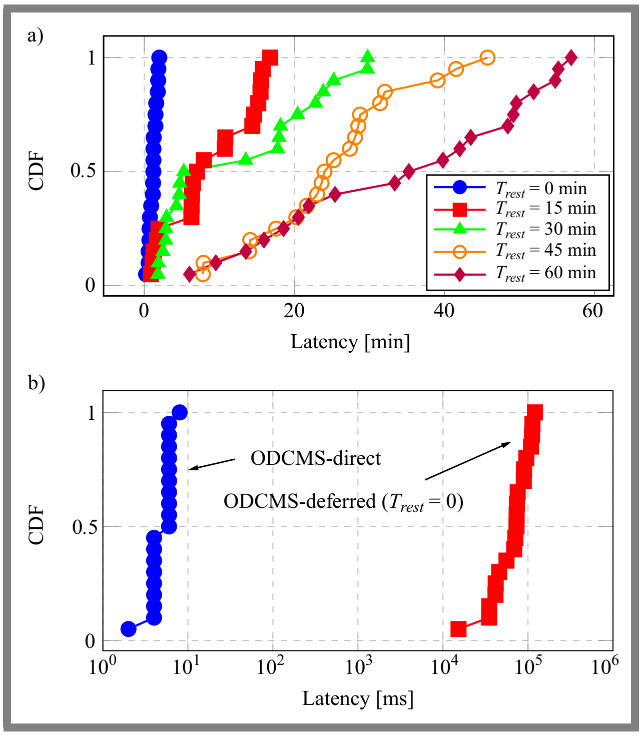

Fig. 10. CDF of latency under ODCMS-deferred for different values of T_{rest} a) and CDF of latency comparison between ODCMS-direct and ODCMS-deferred ($T_{rest} = 0$) b).

deployment scenarios, including obstacle-constrained MS mobility and non-uniform node distributions, as well as multiple MSs and path-length constraints for VP selection and route optimization.

References

[1] K. Gulati *et al.*, "A Review Paper on Wireless Sensor Network Techniques in Internet of Things (IoT)", *Materials Today: Proceedings*, vol. 51, pp. 161–165, 2022 (https://doi.org/10.1016/j.matpr.2021.05.067).

[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, art. no. 14, 2020 (https://doi.org/10.3390/asi3010014).

[3] V. Agarwal, S. Tapaswi, and P. Chanak, "A Survey on Path Planning Techniques for Mobile Sink in IoT-enabled Wireless Sensor Networks", *Wireless Personal Communications*, vol. 119, pp. 211–238, 2021 (https://doi.org/10.1007/s11277-021-08204-w).

[4] N. Benaouda, "Efficient Routing for Delay-energy Tradeoff in Event-based Wireless Sensor Networks", *Journal of Telecommunications and Information Technology*, vol. 98, pp. 69–77, 2024 (https://doi.org/10.26636/jtit.2024.4.1833).

[5] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, "A Survey on Mobility in Wireless Sensor Networks", *Ad Hoc Networks*, vol. 125, art. no. 102726, 2022 (https://doi.org/10.1016/j.adhoc.2021.102726).

[6] S. Jain, R.K. Verma, K.K. Pattanaik, and A. Shukla, "A Survey on Event-driven and Query-driven Hierarchical Routing Protocols for Mobile Sink-based Wireless Sensor Networks", *Journal of Supercomputing*, vol. 78, pp. 11492–11538, 2022 (https://doi.org/10.1007/s11227-022-04327-4).

[7] P.K. Singh and A. Sharma, "An Intelligent WSN-UAV-based IoT Framework for Precision Agriculture Application", *Computers and Electrical Engineering*, vol. 100, art. no. 107912, 2022 (https://doi.org/10.1016/j.compeleceng.2022.107912).

[8] S.M. Bozorgi *et al.*, "An Impressive Clustering Based on the Fuzzy System for UAV-assisted IoT Wireless Networks", *Telecommunication Systems*, vol. 88, art. no. 79, 2025 (https://doi.org/10.1007/s11235-025-01307-w).

[9] M. Xia *et al.*, "WSN Lifetime Maximization: Effects of Energy-sharing and UGV Mobility", *China Communications*, vol. 22, pp. 236–253, 2025 (https://doi.org/10.23919/JCC.fa.2024-0431.202504).

[10] S. Kumari, P.K. Mishra, and V. Anand, "Coverage and Connectivity Aware Deployment Scheme for Autonomous Underwater Vehicles in Underwater Wireless Sensor Networks", *Wireless Personal Communications*, vol. 132, pp. 909–931, 2023 (https://doi.org/10.1007/s11277-023-10642-7).

[11] R. Hamidouche *et al.*, "Classical and Bio-inspired Mobility in Sensor Networks for IoT Applications", *Journal of Network and Computer Applications*, vol. 121, pp. 70–88, 2018 (https://doi.org/10.1016/j.jnca.2018.07.010).

[12] D. Popescu *et al.*, "Advanced UAV-WSN System for Intelligent Monitoring in Precision Agriculture", *Sensors*, vol. 20, art. no. 817, 2020 (https://doi.org/10.3390/s20030817).

[13] K. Medani, C. Gherbi, H. Mabed, and Z. Aliouat, "Energy-efficient Q-learning-based Path Planning for UAV-Aided Data Collection in Agricultural WSNs", *Internet of Things*, vol. 33, art. no. 101698, 2025 (https://doi.org/10.1016/j.iot.2025.101698).

[14] N. Abramov *et al.*, "Intelligent Methods for Forest Fire Detection Using Unmanned Aerial Vehicles", *Fire*, vol. 7, art. no. 89, 2024 (https://doi.org/10.3390/fire7030089).

[15] R.K. Verma and S. Jain, "Energy and Delay Efficient Data Acquisition in Wireless Sensor Networks by Selecting Optimal Visiting Points for Mobile Sink", *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 11671–11684, 2023 (https://doi.org/10.1007/s12652-022-03729-9).

[16] F. Hasan *et al.*, "Energy-efficient Data Collection in Mobile Sink-based Wireless Sensor Networks Using the Hierarchical Clustering Method", *2023 26th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh, 2023 (https://doi.org/10.1109/ICCIT60459.2023.10441076).

[17] A.A. Somasundara, A. Ramamoorthy, and M.B. Srivastava, "Mobile Element Scheduling with Dynamic Deadlines", *IEEE Transactions on Mobile Computing*, vol. 6, pp. 395–410, 2007 (https://doi.org/10.1109/TMC.2007.57).

[18] R. Sugihara and R.K. Gupta, "Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 9, pp. 127–139, 2009 (https://doi.org/10.1109/TMC.2009.113).

[19] B. Yuan, M. Orlowska, and S. Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks", *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 1252–1261, 2007 (https://doi.org/10.1109/TKDE.2007.1062).

[20] A. Mehto, S. Tapaswi, and K.K. Pattanaik, "Virtual Grid-based Rendezvous Point and Sojourn Location Selection for Energy and Delay Efficient Data Acquisition in Wireless Sensor Networks with Mobile Sink", *Wireless Networks*, vol. 26, pp. 3763–3779, 2020 (https://doi.org/10.1007/s11276-020-02293-4).

[21] D.S. Johnson and L.A. McGeoch, "Experimental Analysis of Heuristics for the STSP", in *The Traveling Salesman Problem and Its Variations*, Springer, Boston, MA, pp. 369–443, 2007 (https://doi.org/10.1007/0-306-48213-4_9).

[22] A. Mehto, R.K. Verma, and S. Jain, "Efficient Trajectory Planning and Route Adjustment Strategy for Mobile Sink in WSN-assisted IoT", *2022 IEEE Region 10 Symposium (TENSYMP)*, Mumbai, India, 2022 (https://doi.org/10.1109/TENSYMP54529.2022.9864434).

[23] K. Pitchai, "Determination of Energy Efficient Optimal Sojourn Location of Mobile Sinks in Clustered Corona-based Wireless Sensor Networks", *Peer-to-Peer Networking and Applications*, vol. 15, pp. 1–12, 2022 (https://doi.org/10.1007/s12083-021-01224-0).

[24] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: Energy-aware Path Construction for Data Collection Using Mobile Sink in Wireless Sensor Networks", *IEEE Sensors Journal*, vol. 18, pp. 890–901, 2018 (https://doi.org/10.1109/JSEN.2017.2773119).

[25] X. Fu and X. He, "Energy-balanced Data Collection with Path-constrained Mobile Sink in Wireless Sensor Networks", *AEU - International Journal of Electronics and Communications*, vol. 127, art. no. 153504, 2020 (https://doi.org/10.1016/j.aeue.2020.153504).

[26] Y.M. Raghavendra and U.B. Mahadevaswamy, "Hybrid Rendezvous Clustering Model for Efficient Data Collection in Multi Sink Based Wireless Sensor Networks", *Wireless Personal Communications*, vol. 129, pp. 837–851, 2023 (https://doi.org/10.1007/s11277-022-10158-6).

[27] B. Gutam, P.K. Donta, C. Sekhar, and Y.-C. Hu, "Optimal Rendezvous Points Selection and Mobile Sink Trajectory Construction for Data Collection in WSNs", *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 7147–7158, 2021 (https://doi.org/10.1007/s12652-021-03566-2).

[28] B.R. Al-Kaseem, Z.K. Taha, S.W. Abdulmajeed, and H.S. Al-Raweshidy, "Optimized Energy-efficient Path Planning Strategy in WSN With Multiple Mobile Sinks", *IEEE Access*, vol. 9, pp. 82833–82847, 2021 (https://doi.org/10.1109/ACCESS.2021.3087086).

[29] R. Anwit, A. Tomar, and P.K. Jana, "Scheme for Tour Planning of Mobile Sink in Wireless Sensor Networks", *IET Communications*, vol. 14, pp. 430–439, 2020 (https://doi.org/10.1049/iet-com.2019.0613).

[30] J. Sleegers and D. van den Berg, "Backtracking Algorithms on the Hamiltonian Cycle Problem", *ArXiv*, 2021 (https://doi.org/10.48550/arXiv.2107.00314).

[31] F. Engmann *et al.*, "Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques", *Wireless Communications and Mobile Computing*, vol. 2018, 2018 (https://doi.org/10.1155/2018/8035065).

Nadjib Benaouda, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-4361-9597
E-mail: nadjib.benaouda@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Boubakeur Moussaoui, Ph.D., Associate Professor
Laboratory of Informatics and its Applications of M'sila
https://orcid.org/0000-0003-4991-5044
E-mail: b.moussaoui@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz
Mohammed Boudiaf University, M'sila, Algeria
https://www.univ-msila.dz

Oussama Senouci, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-5345-0713
E-mail: oussama.senouci@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz