Efficient Routing for Delay-energy Tradeoff in Event-based Wireless Sensor Networks

Nadjib Benaouda

University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria

https://doi.org/10.26636/jtit.2024.4.1833

Abstract — Wireless sensor networks (WSNs) play a crucial role in the Internet of Things (IoT) by providing a foundation for collecting, transmitting and processing data from the physical world. Beyond the necessity of proposing solutions that are in line with the constrained resources of sensor nodes, particularly their limited energy capacity, the consideration of real-time data collection becomes essential. This is particularly vital due to the fact that many IoT applications require timely data collection. However, the need to establish energy-efficient routes contradicts the requirement to guarantee timely data collection. Hence, achieving an equilibrium and striking, subsequently, a trade-off between these two issued becomes imperative. To answer this question, a localized delay-bounded and energy-efficient routing protocol (abbreviated as LDER) is presented. It is based on another protocol, namely DEDA, aimed at achieving a higher energy conservation degree. To validate the efficacy of LDER, simulations were conducted using the J-sim simulator. The results demonstrate the ability of LDER to achieve the desired equilibrium and prove its superiority over DEDA.

Keywords — delay-bounded path, energy efficiency, LMST, WSN

1. Introduction

Wireless sensor networks constitute a fundamental component of the IoT ecosystem, playing a pivotal role in facilitating data collection [1], [2]. A WSN consists of a group of nodes that offer perception, processing, and communication capabilities, allowing them to collect and share data through a wireless link. These nodes, commonly referred to as sensor nodes, are deployed in a specific geographic area and measure physical parameters that are relevant to our environment (e.g., temperature, humidity, vibrations, etc.). The nodes relay these measurements to each other, until the data reach the sink, i.e. a specific node that acts as a gateway to the host.

Sensor nodes are low-resource embedded systems, meaning they can only transmit at a limited data rate. Moreover, the capacity of these sensors' batteries limits their operational lifetime. Due to these limitations, new algorithms, methods, and protocols must be developed that are suitable for the specific characteristics of these wireless sensor nodes, with a particular emphasis placed on their limited energy resources [3].

Additionally, many IoT applications require real-time data collection. In fact, the process of monitoring various environmental and physical parameters is crucial for such applications as industrial automation, medical monitoring, and environmental detection, where timely information collection is essential [4]–[6].

Protocols that satisfy those two requirements concurrently are not easy to develop. In fact, lower energy consumption of the sensor nodes contradicts the need to ensure timely arrival of collected data. Opting for lower consumption entails accepting greater data latency, as the established routes would involve numerous nodes transmitting over short distances to conserve energy.

On the other hand, prioritizing lower latency comes at the cost of increased energy consumption. Consequently, it is imperative to establish a trade-off between these two key requirements, namely reduction in energy consumption and timely delivery of the sensed data [7].

To address the challenge mentioned above, we present a localized delay-bounded and energy-efficient routing protocol (LDER). It stems from an existing DEDA protocol [8] which we have enhanced to make it more efficient in preserving energy.

Initially, LDER establishes a sparse topology by applying the local minimum spanning tree (LMST) algorithm to the initial configuration. Then, the shortest-path tree that connects the sink with sensor nodes on the top of this sparse topology is established. Specific events trigger adjustments to the initial routes within the pre-established tree. This adaptation entails incorporating shortcuts from the original network topology when the sensed data cannot be delivered in time or when a more energy-efficient path is possible. In fact, DEDA does not account for this alternative. The protocol fails to provide any mechanisms for identifying paths that might be more energy efficient, other than those stemming from the LMST topology.

To validate the proposal, we simulate both LDER and DEDA using the J-sim network simulator [9]. Various experiments were conducted, producing favorable outcomes that underscore LDER's ability to maintain the desired equilibrium and its advantages over DEDA.

The article is structured as follows. Section 2 provides a concise overview of relevant prior research. Section 3 outlines the preliminary aspects of the proposed approach, including the network model, the definition of the problem, and an introduction to the DEDA protocol. Section 4 presents an overview of LDER. Section 5 shows the details of the proposal. The experimental setup and simulation results are presented in Section 6, followed by conclusions drawn in Section 7.

2. Related Work

The existing literature has seen numerous protocols aimed at tackling the challenge of balancing the delay-energy trade-off in WSNs. In this section, we provide an overview of a selection of these propositions.

In [10], the authors introduce a modified version of the firefly algorithm known as firefly with cyclic randomization (FCR). This variant aims to enable energy-efficient and low-latency WSN routing. The primary focus is on selecting an optimal cluster head that is not only in close proximity to the sink, but also strategically positioned relative to the sensor nodes. This spatial proximity significantly reduces time delays, subsequently leading to an improvement in the data packet transmission speed.

Paper [11] presents a novel algorithm called a delay constrained energy-efficient multi-hop routing algorithm for effective routing in WSNs. The method introduces an innovative approach known as the delay-constrained reliable routing algorithm, minimizing energy consumption by creating efficient clusters without increasing end-to-end delay. The sink node initiates the clustering process, which involves three key steps: cluster head selection, cluster formation, and calculation of the trade-off between energy and delay (TED). Once the clusters are established, the route discovery process uses the depth-first search (DFS) algorithm.

The routing process comprises three stages: link and route cost calculation, end-to-end delay calculation, and data communication. The proposed algorithm offers several advantages, including energy efficiency, enhanced link quality, scalability, improved throughput, reliable data delivery, reduced delay, and cost effectiveness in building clusters and routing.

The authors of [12] introduce a routing solution called multi-QoS constraint multipath routing (MQoScMR) in cluster-based WSNs. The design accommodates multiple quality of service (QoS) constraints. The approach consists of three steps: first, the status of the node is estimated – this phase involves assessing the level of residual energy, delay, and congestion of a given node. In the second step, cluster heads (CH) are selected based on such factors as residual energy, estimated delay and congestion observed during the previous data transmission round. The third step involves the construction of multiple pathways. This requires that potential routing paths be established from the source node to the sink, taking into account the estimate of link cost based on energy level, delay, and congestion indices of the intermediate nodes. Following the completion of this final step, diverse routes are delineated to serve different purposes, in alignment with the specific requirements of a given application.

In [13], the authors present a routing protocol aimed at enhancing both delay and energy efficiency within WSNs by using two-hop information. Their contributions encompass three key aspects. Firstly, they introduce potential relay information (PRI) metrics, which are based on residual energy, distance, and delay factors. This parameter evaluates the link quality of neighboring nodes, ensuring that the subsequent forwarder selected for data packet transmission delivers the packets to the destination with the best quality of service ensured. Secondly, they design a preemptive neighborhood state index (NSI) algorithm enabling the nodes to look two hops ahead in their routing trajectory. This foresight helps make informed forwarding decisions, minimize delays, and distribute traffic loads evenly across the system. Finally, they implement a proactive feedback mechanism to streamline the two-hop information update process.

In [14], a data fusion algorithm is presented that relies on a hybrid delay-sensitive clustering approach. This method integrates the strengths of both single-layer and multi-layer cluster structures. Using a decision function, the proposal dynamically selects the appropriate clustering patterns for the clusters. This dynamic selection process aims to strike a balance between network delay and energy consumption.

The authors of [7] introduce a data aggregation routing technique that leverages the ant-colony optimization metaheuristic. This approach aims to design a routing structure that optimizes overlapping routes while minimizing cumulative transmission power. At the same time, the method ensures that the paths adhere to specified delay limits.

Research efforts focusing on wireless sensor and actor networks (WSANs) offer promising solutions to address the delay-energy tradeoff challenge. Paper [15] introduces a routing protocol that aims to deliver good quality of service (QoS) within the context of a WSAN. The protocol emphasizes such factors as delay and energy consumption. The network is organized in clusters supervised by cluster heads that are selected based on key metrics. These metrics comprise energy capability, connectivity richness, and accessibility of all actors. Furthermore, they tackle the intricacies of sensor-actor communication by introducing an on-demand routing-based data communication protocol that provides access to actor nodes with minimal delay and reduced energy consumption.

In [16], the cluster-based coordination protocol (CCR) is proposed, which organizes sensor nodes into clusters. The CH of each of these groups is linked to the actor which is capable of covering a given geographical area. If several actors cover this area, the closest actor node to this CH is selected. CCR takes advantage of data aggregation by allowing each CH to aggregate its members' data. However, this aggregation process must be executed without compromising the timely delivery of sensed data.

The authors of [17] present a reactive protocol that facilitates the segmentation of sensor nodes within the event area into distinct clusters. The authors achieved this partitioning by pairing each event-detecting sensor node with the nearest actor. Each cluster establishes a data aggregation tree, with its root at the respective actor. This tree is then adjusted based on the actor's feedback, defined by the observed reliability value. This value is defined as the ratio between the number of packets received in a given time and the total number of packets generated within a given interval.

Following this feedback, when the observed reliability value falls below a certain threshold, each sensor selects the neighbor that is geographically closest to the final destination. This helps minimize the number of hops leading to the

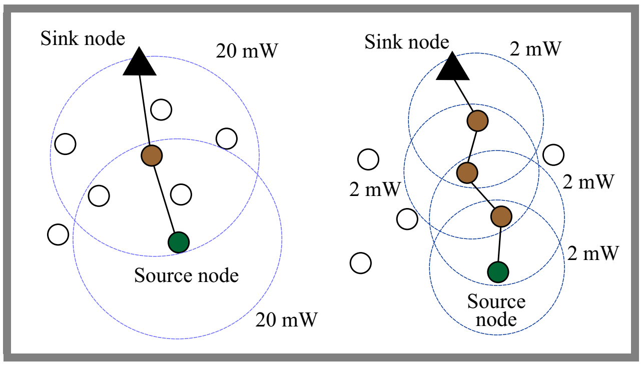

Fig. 1. Two potential routes between the source node and the sink: a) one with a total transmission power of 40 mW and a hop count of 2, and b) one with a power of 8 mW and 4 hops.

destination, thus reducing data latency. However, this minimization comes at the expense of energy consumption. In contrast, when the observed reliability value exceeds the predetermined threshold, indicating excessive reliability, each sensor node forwards its data to its closest neighbor, using a lower transmission power.

3. Preliminaries

This section outlines the preliminary aspects of the proposed approach, elucidates the network model, defines the problem, and presents the DEDA protocol. The network model adopted in this study is outlined as follows:

- all nodes within the network are stationary,
- each sensor i can change its communication range by adjusting its transmission power level,
- any sensor can reach a maximum transmission power of P_{max}, which is equivalent to a maximum range of R_{max}.

The topology of the network is modeled by the $G = (V, E)$ graph, where: $V = (v1, v2, \ldots, v_n)$ comprises a collection of vertices that represent the nodes within the network and E is the set of arcs (links) that interconnect these nodes. The latter set is defined as follows:

$$E = \{(u, v) | u \in V, \quad v \in V, \quad \delta_{u,v} \leqslant R_{max}\} \,. \tag{1}$$

Considering that $\delta_{u,v} = \sqrt{(x_u - x_v)^2 + (y_x - y_v)^2}$ corresponds to the Euclidean distance between the two nodes u and v, where $u = (x_u; y_u)$ and $v = (x_v; y_v)$, the following formula calculates the transmission power required for communication between node i and its neighboring node j [18]:

$$Power_{i,j} = \delta_{i,j}^{\alpha} \,, \tag{2}$$

where $\delta_{i,j}$ represents the distance that separates i from j, which corresponds to the transmission range used by node i to communicate with j, α is the path loss exponent that is typically assigned a value of 2 or 4 [19]. Thus, the power values are expressed as a function of the transmission range (i.e., distance) without providing power levels in a specific energy unit, such as Joules, to allow to compare the protocol with other approaches.

3.1. Problem Definition

In this article, the primary objective is to delve into the intricacies of the routing challenge, specifically with respect to the efficient transmission of the collected data from individual sensor nodes to the designated sink node within the WSN. The limited energy source necessitates energy efficient communication. At the same time, the transmission of collected data must not exceed a specified latency level. Balancing these two objectives poses a challenge: minimizing energy consumption during the routing process necessitates considering paths with short links, thus resulting in a high number of hops and increased data latency.

Figure 1, adapted from [20], depicts two potential routes connecting the source and the sink. Figure 1a shows a two-hop route with a total transmission power of 40 mW, and Fig. 1b shows a second, more energy-efficient route with a total transmission power of 8 mW, but a total number of hops of four.

This illustrates two conflicting objectives: to minimize the total transmission power while simultaneously ensuring that communications adhere to the imposed temporal constraint. This is precisely the challenge that we aim to address in this research.

3.2. DEDA Protocol

In this section, we present the DEDA protocol [8], serving as the foundation of our proposal.

DEDA aims to build aggregation trees with energy-efficient and time-bounded paths. An actor (or sink) serves as the root of each tree, which then connects to its corresponding sensor nodes – Fig. 2. DEDA first establishes a new sparse topology over the original network graph based on the LMST algorithm. On top of this new topology, it builds a tree that connects the actor with its associated sensors, Fig. 2b. This tree is used, without modification, if the given deadline is not exceeded. Otherwise, this tree is adjusted by borrowing other links from the original graph. This adjustment is made on the basis of a certain value of DEP which reflects the desired progress on establishing the sparse topology, ensuring that the deadline is met. This value is calculated as follows:

$$DEP = \left\lceil \frac{LD(w)}{TL - MED} \right\rceil \times l(w) \,, \tag{3}$$

where $LD(w)$ denotes the distance between the current node w and the actor in the sparse topology. The time limit (TL) represents the user-specified deadline. MED is the highest experienced delay specified in all reports received from w children. The value of $l(w)$ is determined based on the traffic load within the network. It assumes the value of 1 when traffic intensity is relatively low. In such scenarios, the distance between two nodes corresponds to the number of hops along the path connecting them.

However, if traffic intensity is high, $l(w)$ corresponds to the degree of node w, and the distance between two nodes becomes the cumulative sum of degrees between all nodes within the connecting path. In the latter scenario, the distance

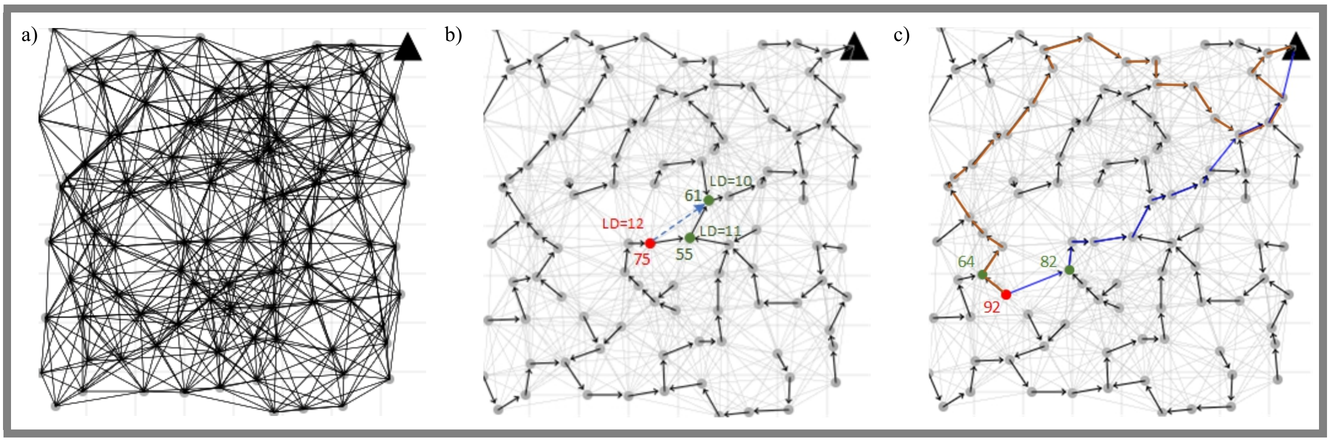

Fig. 2. DEDA principle and the solution addressed in this work: a) the initial topology generated using the maximum transmission power of the network nodes, b) the shortest path tree built on the local minimum spanning tree (LMST) of the initial topology, and c) the route of the source node 92, assuming a time limit equal to ∞. The orange line marks the route built with DEDA, while the blue line represents the outcome of the proposed solution.

is computed using this approach due to the impact of the number of nodes contending for access to the wireless channel on the basis of data latency.

To elaborate on the fundamental concept behind DEDA, let us consider the example shown in Fig. 2b. Suppose we calculate the distance in hops and set TL to 11. Then, node 75 cannot route its data through its parent (node 55) in the pre-established aggregation tree because this last node cannot guarantee the prompt delivery of these data, i.e. $LD(75) = 12$. Consequently, node 75 alters its parent and opts for its neighbor 61 from the original topology. This selection is based on the fact that neighbor 61 facilitates a progress of 2 hops within the sparse topology, that is, $LD(75) - LD(61) = 12 - 10 = 2$, equivalent to the value $DEP = \lceil \frac{12}{11-0} \rceil \times 1 = 2$. Here, MED used in the DEP calculation example equals 0, as we assumed, for illustrative purposes, that node 75 is the source node. Therefore, it does not forward any reports received from its children, but is the one that initiates data transmission.

3.3. Limitations of the DEDA Protocol

DEDA was purposefully designed to gather data from all nodes linked to a particular actor in response to the last query. Here, every node transmits its data which are subsequently aggregated across various paths within the established aggregation tree. These data traverse energy-efficient routes, since these paths were constructed based on the LMST topology.

However, when only the sensor nodes that detect events transmit their data to the actors, DEDA encounters inefficiencies. Consider the scenario shown in Fig. 2c, where source node 92 (shown in red) aims to transmit its data to its corresponding actor. Assuming an infinite delay bound, node 92 would employ its path within the aggregation tree (indicated by the orange route in Fig. 2c) without any alterations. This final route is notably lengthy, resulting in a considerable energy expenditure, even if it is made exclusively of short links, i.e., nodes utilizing minimal transmission power.

Under these circumstances, opting for alternative paths with reduced energy expenditures is better. For instance, it would be more advantageous for node 92 to route its data through its neighbor 82, utilizing link $e_{92,82}$ from the original topology, as depicted in Fig. 2c.

The question that arises here is how these types of neighbors can be identified and how to efficiently introduce shortcuts from the initial topology to yield paths with reduced energy expenditures. In fact, along this trajectory that our contribution is made, enhancing DEDA's capabilities to find shortcuts by incorporating links from the original topology not only to meet the specified time constraints but also to establish paths with decreased energy expenditures.

4. Overview of the LDER Protocol

In this section, we present an overview of our proposed LDER protocol, designed to enhance DEDA capabilities. LDER operates through a sequence of four steps:

1) Initially, each node in the network gathers data concerning its neighboring nodes, specifically their locations and identifiers. This collection is characterized by the hello messages that each node sends to its vicinity.
2) Based on this information, a new sparse topology is constructed on top of the initial network topology, using the LMST algorithm [21]. The use of such a sparse topology has the benefit of working with an energy-efficient topology from the beginning, as it preserves only the links that require minimal energy consumption.
3) After establishing this new sparse topology, the shortest path tree is derived over this LMST topology, using the sink as the root. We established this tree based on the number of hops along the traversed routes. During this step, each node determines its initial parent and collects information pertinent to all of its neighbors, including those that are not included in the set of neighbors in the sparse topology. Each node specifically obtains information on

the distance and cost of the optimum path connecting each neighbor within the initial topology to the sink. This knowledge enables each node to create shortcuts within the initial topology and choose its new parent.

4) Upon the creation of this tree, the routing of data from the source node to the sink can be performed whenever an event is detected. Each node transmits the messages it receives to its designated parent node, established in the preceding phase if this parent guarantees the prompt delivery of these messages and no other neighbor exists, among all its neighbors in the initial topology, that also ensures timely delivery while simultaneously providing a more energy-efficient route to the sink. Alternatively, a shortcut in the original topology can be added by selecting a new parent node from the neighbors of the current node within this initial topology.

5. Details of LDER

This section highlights the details of LDER protocol by elaborating the specifics of each step outlined in the preceding section. The main data structures employed at every sensor node i are presented in Tab. 1.

Tab. 1. Main data structures employed at each sensor node i.

Data	Description
ID_i	ID of sensor node i
$SinkID_i$	ID of the sink
$Coordinate_i$	Coordinates of node i
$Parent_i$	ID of parent of node i
LD_i	Number of hops along optimal path linking node i with the sink within the LMST topology
TL	Time limit
N_i^{LMST}	Set of nodes neighboring node i within the sparse topology (LMST)
N_i^{MAX}	Denotes the collection of neighboring nodes for node i within the original topology. Every element within this set comprises the neighbor's ID, the path distance (measured in hops) connecting the said neighbor with the sink within the sparse topology, and lastly, the energy expenditure associated with this path

5.1. Collecting Neighboring Node Information

In the initial phase, every sensor node initiates the process of gathering information regarding its neighboring nodes. To do this, each node generates a hello message containing its ID and coordinates, which is then transmitted using its maximum transmission power. This data collection phase continues for a duration of t_{hello}, during which each node transmits its message and awaits acknowledgment from its neighboring nodes.

5.2. Sparse Topology Formation

Once the t_{hello} interval has elapsed, which signifies the end of the phase during which information regarding the neighbors of each sensor node in the initial topology is collected, the LMST algorithm is launched. This phase is essential, as it establishes the network's new topology based on energy-efficient connections between nodes. The entire procedure works as follows [21]:

- Computation of local MST. Each sensor node uses the coordinates of its neighboring nodes to calculate a local minimal spanning tree (MST).
- Selection of new neighbors. Once the MST has been calculated, each node identifies its new neighbors within the new topology. This is achieved by selecting the nodes that are one hop away within the newly established MST.
- Transformation into an undirected graph. The preliminary result of this MST-based neighbor selection procedure has the form of a directed graph. To establish a functional and resilient topology, the directed graph must be converted to an undirected graph. This can be accomplished by relying on one of two methods:
  - LMST^{-} method. An edge between nodes i and j is established in the new topology only if both directed edges $e_{i,j}$ and $e_{j,i}$ are present in the respective MSTs of nodes i and j. This ensures a mutual agreement for the connection, leading to a sparse but reliable topology. This variant is known as LMST^{-}.
  - LMST^{+} method. Alternatively, an edge between nodes i and j is created if either of the directed edges $e_{i,j}$ or $e_{j,i}$ is present in the MSTs. This more relaxed criterion results in a denser topology, referred to as LMST^{+}.

Here, sparse topology was formed using Prim's algorithm, while the weights of the links were calculated using Eq. (2). As a variation of LMST, we have adopted LMST^{-}. This was done by allowing each node to share information about its newly computed neighbors using route discovery messages while constructing the shortest path tree.

5.3. Calculation of the Shortest Path Tree Across the Sparse Topology

When the formation of the sparse topology is complete, the process of creating the shortest path tree connecting sensor nodes to the sink may commence. It is initiated, as depicted in line 1 of Algorithm 1, by the sink broadcasting a route discovery (RD) message to its neighboring nodes. This message comprises several fields:

- Sender,
- Sink ID,
- LD denoting the distance (in terms of hops) traveled by the RD message within the sparse topology (LMST) since its dissemination by the sink (initialized to 0),
- TL representing the designated time limit,

Algorithm 1 The process of determining the shortest path tree across the sparse topology.

1: The sink creates and broadcasts a RD message
 ▷ R_{RD} is the set of nodes receiving the RD message
2: **for** each node $i \in R_{RD}$ **do**
3: Node i updates the element that matches $RD.sender$ in N_i^{MAX} with $RD.LD$ and $RD.cost$
4: **if** $ID_i \notin RD.N_{LMST}$ and $RD.sender \in N_i^{LMST}$ **then**
5: Node i deletes $RD.sender$ from its set N_i^{LMST}
6: **end if**
7: **if** $LD_i > RD.LD + 1$ and $RD.sender \in N_i^{LMST}$ **then**
8: $Parent_i \leftarrow RD.sender$
9: $LD_i \leftarrow RD.LD + 1$
10: $SinkID_i \leftarrow RD.sinkID$
11: $TL \leftarrow RD.TL$
12: $RD.cost \leftarrow RD.cost + power_{i,RD.sender}$
13: $RD.sender \leftarrow ID_i$
14: $RD.LD \leftarrow LD_i$
15: Node i re-broadcasts the RD message
16: **end if**
17: **end for**

- Cost signifying the cumulative cost since the initial broadcast (initialized to 0),
- N_{LMST} which indicates the set of neighbors of the sender within the sparse topology.

Upon receiving this message, each node i proceeds to update the entry in its set N_i^{MAX} that corresponds to RD.sender with RD.cost and RD.LD values. Taking into account the use of the LMST$^-$ variation, node i additionally removes RD.sender from its set N_i^{MAX} under the condition that the mentioned sender is part of N_i^{MAX} and ID_i is not included in the $RD.N_{LMST}$ set. Subsequently, node i verifies whether the RD message comes from one of its LMST neighbor with a better path. If this is the case, it proceeds to update its internal data along with the fields of the RD message, as outlined in lines 8–11 and 12–14 of Algorithm 1, respectively. Following this update, the node re-broadcasts the RD message with maximum transmission power.

Concluding this phase, each node i acquires knowledge about the distance and cost of the optimal path that links each neighbor within the N_i^{MAX} set with the sink. This information serves as a foundation for the subsequent phase.

5.4. Determining an Appropriate Route in Response to an Event

Following the creation of the shortest path tree, the sensor nodes await for events to unfold. Upon detection of an event, the source node generates a parent decision message (PDM). The purpose of this message is to determine a route that guarantees a prompt transmission of the sensed data while potentially conserving more energy than the initial path established in the preceding phase. Therefore, each node aiming to determine the next hop for the PDM message evaluates whether its previously designated primary parent remains the optimal choice.

Tab. 2. Simulation parameters.

Parameter	Value
Number of sensor nodes	500
Sink position	Center of the field
Number of source nodes	4
Density	20
Communication range	80 m
Path loss α	2

In other words, this parent should guarantee data arrival prior to the set deadline, while no other neighboring node in N_i^{MAX} suggests a path with a superior energy efficiency level. If these criteria cannot be met, exploration of potential shortcuts within the original topology becomes imperative.

In a more detailed explanation, each node i, aiming to identify the next hop for the PDM message, starts by calculating the desired progress value (DEP). The latter is defined as the desired progress, measured in the count of hops, of the PDM message within the sparse LMST topology, so as not to exceed the specified time limit. This DEP value is calculated as follows:

$$DEP = \left\lceil \frac{LD_i}{TL - ED} \right\rceil , \tag{4}$$

In this formula, ED corresponds to the experienced delay of the PDM message from the moment of its dissemination by the source node.

Upon the calculation of DEP, node i proceeds to compute, for every neighbor j within N_i^{MAX}, the proposed progress that j offers, using the following formula:

$$Progress_j = LD_i - LD_j , \tag{5}$$

where LD_j is obtained from the entry corresponding to node j within set N_i^{MAX}.

After calculating these values, node i selects its parent n as:

$$n = \begin{cases} \underset{j \in N_i^{MAX}}{\arg\min}\ Cost_j & , \text{ if } \exists\, j \in N_i^{MAX} \text{ such that } Progress_j \geqslant DEP \\ \underset{j \in N_i^{MAX}}{\arg\max}\ Progress_j , & \text{otherwise} \end{cases} . \tag{6}$$

In order to ensure a progress rate that exceeds DEP, node i chooses, as its parent node n, from among all of its neighbors, the one which presents the lowest energy cost towards the sink. In cases where no neighbor offers an advancement greater than DEP, node i opts for the neighbor that provides the most substantial advancement.

Once this node n has been selected, node i updates the fields of the message (PDM) and sends it to n. The same process as illustrated in Algorithm 2 is repeated until the PDM message successfully reaches the sink. Subsequent data will then travel an identical route as undertaken by the PDM message.

Algorithm 2 Routing process of a PDM message.

▷ s is the source node
▷ c is the current node of the PDM message
▷ n corresponds to the selected parent

```
c ← s
while c ≠ SinkID do
    Node c computes DEP according to Eq. (4)
    for each node j ∈ N_i^MAX do
        Node c computes Progress_j using Eq. (5)
    end for
    Node c selects n according to Eq. (6)
    c ← n
end while
```

6. Experiments and Results

We conducted the simulation using the J-sim simulator [9], incorporating both our protocol and DEDA, for evaluation purposes. Table 2 presents the primary simulation parameters that were utilized. The area of the deployment field was calculated as follows: network density d is defined by the number of nodes per unit area, meaning that there are d nodes inside the communication region of each node, represented as a circle of radius r and area πr^2.

To equally distribute n nodes across a deployment region with a uniform density d, the total deployment area A must be sufficient to support all n nodes at that density. Consequently, the necessary area is determined by the $\frac{(n\pi r)^2}{d}$ formula. In a square deployment region, side length L is the square root of the area, resulting in:

$$L = \sqrt{\frac{(n\pi r^2)}{d}} . \tag{7}$$

Therefore, a square deployment area is adopted in the experiments, with a side length of L calculated with the use of the previous formula.

The primary goal of the experiments conducted is two-fold. Firstly, it aims to demonstrate the ability of LDER to achieve the intended trade-off, and secondly, to show its proficiency in discovering paths with lower energy costs compared to those computed by DEDA. To achieve this goal, a series of various experiments were undertaken, considering various time limits Γ. The results obtained are presented in the form of snapshots that are used to visually depict the routes established by each protocol for various values of Γ. These snapshots distinctly illustrate the introduction of various shortcuts.

The curves show the different costs (i.e., total transmission powers) of the routes built by each protocol according to Γ.

Figure 3 represents the distinct routes obtained following the execution of the proposed LDER protocol with corresponding Γ values of 6, 8, 10, and ∞. Firstly, it is evident that the hop count for each route consistently remains below the specified Γ value. Secondly, there is a noticeable trend – the average hop count for the established routes gradually increases as the Γ value grows.

This phenomenon is logical considering a closer examination of Fig. 3, where a decrease in Γ is associated with an increase in the number of shortcuts. This stems from the necessity to continually locate nodes capable of ensuring progress within the initial topology while adhering to the designated time limit. In fact, this decrease in the average hop count is coupled with a decrease in energy preservation. This stems from the incorporation of shortcuts, which entails the inclusion of less energy-efficient links. This addition becomes more pronounced with each decrease of Γ.

Figure 4 presents a visualization of the various routes obtained after running DEDA using similar Γ values. The same

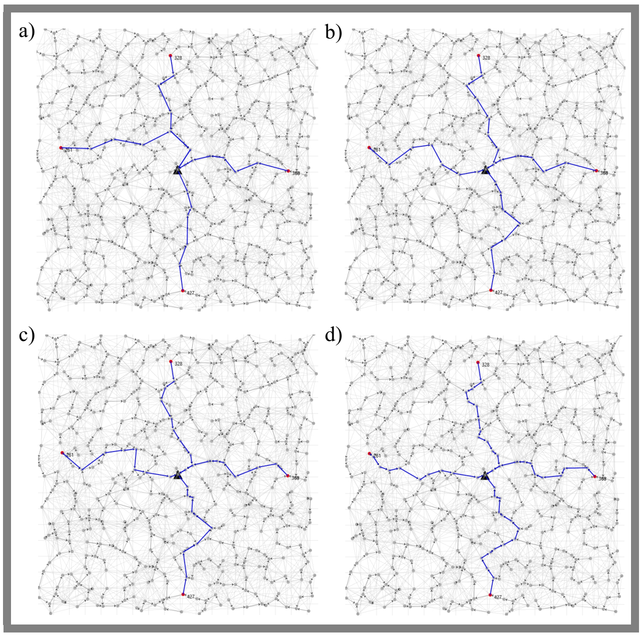

Fig. 3. Snapshots showing the distinct routes constructed by LDER using: a) Γ=6, b) Γ=8, c) Γ=10, and d) $\Gamma=\infty$ hops.

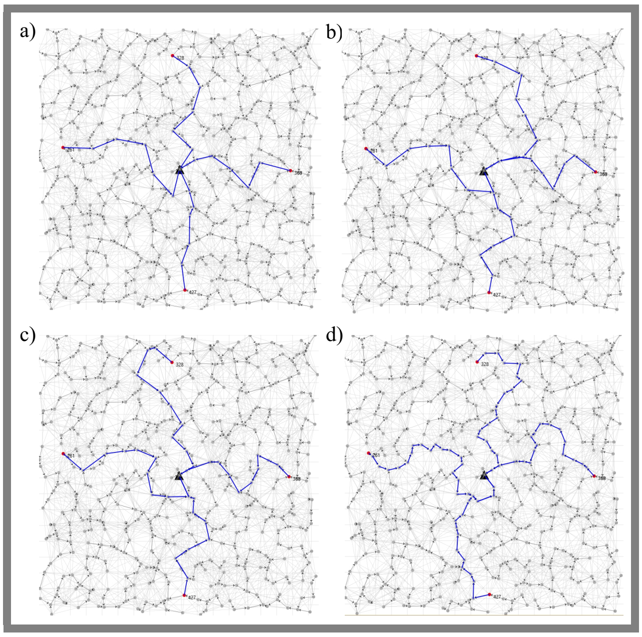

Fig. 4. Snapshots displaying different routes constructed by DEDA using: a) Γ=6, b) Γ=8, c) Γ=10, and d) $\Gamma=\infty$ hops.

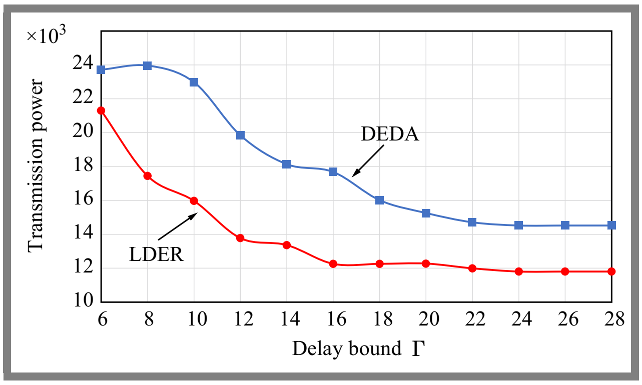

Fig. 5. Average total transmission power of the routes formed by LDER and DEDA with respect to Γ.

observations apply. However, when comparing Fig. 3d with Fig. 4d, it becomes apparent that DEDA does not incorporate any shortcuts. The established routes are entirely restricted to the LMST topology due to the large Γ value, eliminating the need to include shortcuts from the initial topology.

On the contrary, the LDER protocol follows a different approach, where specific shortcuts are introduced. These shortcuts are not primarily aimed at ensuring timely data delivery but are rather aimed at adopting paths that are more energy efficient than those provided by the LMST topology.

Figure 5 shows the ability of LDER to discover routes with lower total transmission power compared to the routes formed by DEDA. In this experiment, we compared the total transmission power of the tree routing structures constructed by DEDA and LDER across varying time limits. The results reveal a decent improvement in power efficiency with the adoption of LDER. For example, at a time limit of 8, LDER demonstrated a 27.13% reduction in transmission power compared to DEDA.

As the time limit increased, the benefits persisted, with LDER consistently outperforming DEDA by significant margins. These findings underscore the effectiveness of LDER in enhancing the energy conservation aspects of the routing process, highlighting its potential impact on the overall performance of wireless sensor networks.

7. Conclusion

This study has focused on energy-efficient and real-time data routing processes within event-based wireless sensor networks, a critical domain within the realm of the IoT. We introduced the LDER protocol which adeptly balances the trade-off between minimizing energy consumption and ensuring prompt delivery of sensed data. The novel approach is based upon the foundation of another protocol, known as DEDA.

To validate the effectiveness of LDER, we conducted comprehensive simulations using the J-sim simulator. We seamlessly integrated both LDER and DEDA into this simulator. Through a series of experiments, we obtained positive outcomes that underscore the performance of the protocol.

References

[1] A.M.K. Abdulzahra, A.K.M. Al-Qurabat, and S.A. Abdulzahra, "Optimizing Energy Consumption in WSN-based IoT Using Unequal Clustering and Sleep Scheduling Methods", *Internet of Things*, vol. 22, art. no. 100765, 2023 (https://doi.org/10.1016/j.iot.2023.100765).

[2] T. Fang and Y. Yang, "Distributed Communication Protocol in Wireless Sensor Network Based on Internet of Things Technology", *Wireless Personal Communications*, vol. 126, no. 3, pp. 2361–2377, 2022 (https://doi.org/10.1007/s11277-021-09203-7).

[3] B. A. Begum and S. V. Nandury, "Data Aggregation Protocols for WSN and IoT Applications – A Comprehensive Survey", *Journal of King Saud University – Computer and Information Sciences*, vol. 35, no. 2, pp. 651–681, 2023 (https://doi.org/10.1016/j.jksuci.2023.01.008).

[4] J. Bian *et al.*, "Machine Learning in Real-time Internet of Things (IoT) Systems: A Survey", *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8364–8386, 2022 (https://doi.org/10.1016/10.1109/JIOT.2022.3161050).

[5] R. Kavra, A. Gupta, and S. Kansal, "Optimization of Energy and Delay on Interval Data Based Graph Model of Wireless Sensor Networks", *Wireless Networks*, vol. 29, no. 5, pp. 2293–2311, 2023 (https://doi.org/10.1007/s11276-023-03292-x).

[6] A. Hassan, A. Anter, and M. Kayed, "A Survey on Extending the Lifetime for Wireless Sensor Networks in Real-time Applications", *International Journal of Wireless Information Networks*, vol. 28, no. 1, pp. 77–103, 2021 (https://doi.org/10.1007/s10776-020-00502-7).

[7] N. Benaouda and A. Lahlouhi, "Ant-based Delay-bounded and Power-efficient Data Aggregation in Wireless Sensor Networks", *International Journal of Pervasive Computing and Communications*, vol. 15, no. 2, pp. 97–119, 2019 (https://doi.org/10.1108/IJPCC-04-2019-0037).

[8] X. Li *et al.*, "Localized Delay-bounded and Energy-efficient Data Aggregation in Wireless Sensor and Actor Networks", *Wireless Communications and Mobile Computing*, vol. 11, no. 12, pp. 1603–1617, 2011 (https://doi.org/10.1002/wcm.1222).

[9] A. Sobeih *et al.*, "J-sim: a Simulation and Emulation Environment for Wireless Sensor Networks", *IEEE Wireless Communications*, vol. 13, no. 4, pp. 104–119, 2006 (https://doi.org/10.1109/MWC.2006.1678171).

[10] A. Sarkar and T.S. Murugan, "Cluster Head Selection for Energy Efficient and Delay-less Routing in Wireless Sensor Network", *Wireless Networks*, vol. 25, no. 1, pp. 303–320, 2019 (https://doi.org/10.1007/s11276-017-1558-2).

[11] M. Selvi *et al.*, "A Rule Based Delay Constrained Energy Efficient Routing Technique for Wireless Sensor Networks", *Cluster Computing*, vol. 22, no. 5, pp. 10839–10848, 2019 (https://doi.org/10.1007/s10586-017-1191-y).

[12] J. Agarkhed, P.Y. Dattatraya, and S. Patil, "Multi-QoS Constraint Multipath Routing in Cluster-based Wireless Sensor Network", *International Journal of Information Technology*, vol. 13, no. 3, pp. 865–876, 2021 (https://doi.org/10.1007/s41870-020-00461-5).

[13] E.D. Tita, W.-P. Nwadiugwu, J.M. Lee, and D.-S. Kim, "Real-time Optimizations in Energy Profiles and End-to-end Delay in WSN Using Two-hop Information", *Computer Communications*, vol. 172, pp. 169–182, 2021 (https://doi.org/10.1016/j.comcom.2021.02.007).

[14] X. Liu *et al.*, "Intelligent Data Fusion Algorithm Based on Hybrid Delay-aware Adaptive Clustering in Wireless Sensor Networks", *Future Generation Computer Systems*, vol. 104, pp. 1–14, 2020 (https://doi.org/10.1016/j.future.2019.10.001).

[15] S. Yahiaoui *et al.*, "An Energy Efficient and QoS Aware Routing Protocol for Wireless Sensor and Actuator Networks", *AEU – International Journal of Electronics and Communications*, vol. 83, pp. 193–203, 2018 (https://doi.org/10.1016/j.aeue.2017.08.045).

[16] G. Shah, M. Bozyigit, and F. Hussain, "Cluster-based Coordination and Routing Framework for Wireless Sensor and Actor Networks",

Wireless Communications and Mobile Computing, vol. 11, pp. 1140–1154, 2011 (https://doi.org/10.1002/wcm.885).

[17] T. Melodia, D. Pompili, V.C. Gungor, and I.F. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks", *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1116–1129, 2007 (https://doi.org/10.1109/TMC.2007.1009).

[18] H. Bagci, I. Korpeoglu, and A. Yazıcı, "A Distributed Fault-tolerant Topology Control Algorithm for Heterogeneous Wireless Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 914–923, 2015 (https://doi.org/10.1109/TPDS.2014.2316142).

[19] 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).

[20] K. Li and C.-C. Shen, "Balancing Transmission Power and Hop Count in ad hoc Unicast Routing with Swarm Intelligence", *2008 IEEE Swarm Intelligence Symposium, SIS 2008*, St. Louis, USA, 2008, (https://doi.org/10.1109/SIS.2008.4668323).

[21] N. Li, J.C. Hou, and L. Sha, "Design and Analysis of an MST-based Topology Control Algorithm", *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005 (https://doi.org/10.1109/TWC.2005.846971).

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