EGAL: Enhancing LoRa Network Lifetime with Load Balancing

Malaika Afra Taj, Kanav Sabharwal, Mun Choon Chan
School of Computing, National University of Singapore, Singapore
mafra@comp.nus.edu.sg, kanav.sabharwal@u.nus.edu, chanmc@comp.nus.edu.sg

***Abstract*—LoRa networks have emerged as a promising solution for long-range, low-power communication in IoT applications. However, a significant challenge in LoRa networks is the uneven battery depletion among nodes, due to the varying transmission configurations required to support different distances between the nodes and the gateway. This disparity in battery consumption poses a substantial challenge to the overall network lifetime. This paper introduces EGAL, a relay-based approach designed to address the battery imbalance issue by ensuring balanced energy consumption across all nodes. Unlike purely multi-hop networks, EGAL follows a hybrid approach that selectively uses relays when necessary while maintaining direct connections to the gateway. EGAL employs a reward-based algorithm that dynamically adjusts relay duties and integrates predictive analytics, thereby enhancing network lifespan with minimal overhead. The proposed solution is validated through NS3 simulation and real-world testbed, demonstrating significant improvements in network lifetime. Specifically, in simulations, EGAL exhibits up to a 457% increase in network lifetime over standard LoRaWAN, and in hardware tests, it shows up to a 70% decrease in total energy consumption. The key contributions of this work include the development of the EGAL algorithm, its integration with standard LoRaWAN, and extensive validation of its effectiveness.**

***Index Terms*—LoRa Networks, Battery imbalance, Network lifetime, Relay-based approach, Balanced energy consumption, Minimal overhead**

I. Introduction

LoRa is a leading Low Power Wide Area Network (LP-WAN) technology developed by Semtech [1], renowned for its long-range communication capabilities and low power consumption [2]. The open-source nature of LoRa, coupled with its operation in the unlicensed spectrum, makes it an ideal choice for a wide array of Internet of Things (IoT) applications, including smart cities, industrial IoT, and environmental monitoring [3]–[6]. Despite its advantages, a persistent challenge in LoRa networks is the imbalance in energy consumption among nodes. In typical deployments, nodes located farther from the gateway must maintain long-range communication by utilizing higher Spreading Factors (SFs)—a configuration that nearly doubles per-packet energy consumption with each increment in SF. As a result, these distant nodes expend significantly more energy, potentially consuming up to a maximum of 32 times more energy than those located closer to the gateway. This energy imbalance causes certain nodes to deplete their batteries much faster than others, thereby reducing the overall network lifetime.

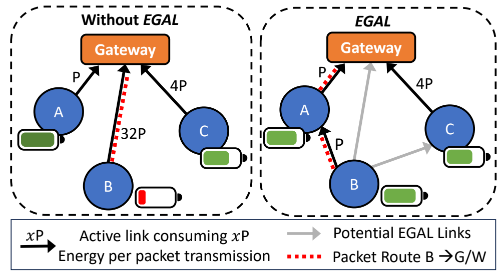


Fig. 1: EGAL balances battery load across the network by dynamically offloading packets from a draining end-node to an intermediate node with sufficient battery capacity.

We define **network lifetime** as the duration from the network's initialization until the battery of the first node is exhausted. This definition is particularly relevant in mission-critical scenarios, such as mine site and underground tunnel monitoring. In these inaccessible environments, frequent battery replacements are impractical, and overall network maintenance costs can be significantly reduced if all nodes sustain their operations for as long as possible and reach battery depletion simultaneously. Moreover, in such scenarios, the failure of even a single node can partition the network and impact its performance. While adding redundant nodes could extend the network's lifetime, this approach also increases operational costs. Therefore, to effectively extend network lifetime, it is essential to balance energy consumption across all nodes to prevent premature battery depletion in some nodes.

In the past, various efforts have been made to improve energy efficiency in LoRa networks. Non-relay-based methods [7]–[12], which involve direct communication between nodes and the gateway, focus on identifying the most energy-efficient configurations for communication. However, these methods fail to address the fundamental issue of imbalance energy consumption across the LoRa network.

Conversely, relay-based approaches[1] that rely on static scheduling [13] pre-plan fixed relay routes for each node. Such strategy inherently limits the responsiveness to changing network conditions. Dynamic strategies attempt to overcome this limitation by making global, centralized decisions [14] that are updated according to current network conditions. However,

[1]A relay refers to an active LoRa node that forwards data from other nodes in addition to transmitting its own data.

this approach incurs high overheads to continuously update the central controller. To mitigate this overhead, decision-making can be delegated to individual nodes [15]. Yet, without a comprehensive view of the entire network, these local decisions tend to be greedy and globally sub-optimal.

To address the limitations in balancing energy consumption in LoRa networks, we propose EGAL[2], a dynamic relay-based approach with centralized decision-making. As illustrated in Figure 1, EGAL offloads data transmission to intermediate nodes with higher battery levels when required, addressing the battery disparity problem. Instead of purely operating as a multi-hop LoRa network, EGAL focuses on using relays to complement a standard single-hop LoRa network. By dynamically adjusting relay duties, EGAL aims to maintain even battery levels across all nodes, thereby enhancing network lifetime.

The challenges in developing a system to effectively address the problem of energy imbalance in LoRa networks can be summarized as follows:

1. Strategic Relay Selection: Relay selection should dynamically adapt to changing network conditions. The process should balance the use of relay nodes to minimize energy consumption while determining when direct communication with the gateway is more appropriate, ensuring that no single node is overburdened with relay requests.

2. Minimizing Overhead: While centralized decision-making with global view of the network allows effective relay selection, the performance gain can be negated if there are frequent control updates to the gateway resulting in substantial overhead. It is crucial to design a protocol that minimizes this overhead while retaining a sufficiently "global" view.

3. Optimizing Link Exploration: Another challenge in maintaining a centralized view of the network is the need to explore inactive[3] network links. While exploring new routes may uncover more energy-efficient options, this process can incur substantial energy on exploring sub-optimal choices. The protocol must strike a balance between efficiently using known routes and exploring new ones, adapting to evolving network conditions without incurring surplus energy costs.

EGAL addresses the challenges discussed above as follows. First, EGAL employs a reward metric for every communication link, reflecting the network load faced by the involved nodes and their residual battery levels. By selecting the link with the highest reward, the algorithm ensures that data is transmitted via the *optimal path*, addressing the first challenge of strategic relay selection.

The use of a reward metric also helps quantify multiple parameters that evaluate the cost of a communication link into a single numerical value. This compression significantly reduces the amount of information that needs to be exchanged within the network, resulting in minimal overhead. EGAL therefore effectively *minimizes overhead* while maintaining centralized decision logic.

Finally, EGAL addresses the challenge of optimizing link exploration by estimating the rewards for inactive links. This estimation is done at the gateway with zero additional overhead, allowing the system to maintain an up-to-date understanding of the potential value of each link, *eliminating the need for active exploration.*

In summary, **the key contributions** of our work are:

- We present EGAL, a relay-based centralized approach that calculates rewards for active communication links at the nodes and estimates rewards for inactive links at the gateway. This method ensures balanced energy usage across all nodes with minimal overhead, thereby extending the network's lifetime.
- We implement a comprehensive system that seamlessly integrates EGAL with the standard LoRaWAN stack, enabling easy deployment in existing LoRa networks.
- We extensively validate our implementation using NS3 simulations and real-world deployment.

II. Background and Energy Framework

A. LoRa - An Overview:

LoRa (Long Range) is a physical layer technology operating in sub-GHz unlicensed frequency bands, offering communication ranges up to tens of kilometers using Chirp Spread Spectrum (CSS) modulation [16]. A key feature of LoRa is the spreading factor (SF), a configurable transmission parameter that determines how many data bits are encoded in each chirp. SF values range from 7, which offers shorter symbol duration and lower range, to 12, which provides longer range with increased latency.

Time on Air (ToA) refers to the time required to transmit a packet. The following equations are used for calculating LoRa's airtime:[4]

$$\mathrm{T_{symbol}} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{1}$$

$$\mathrm{ToA} = (12.15 + \mathrm{N_{payload}}) \times \mathrm{T_{symbol}} \tag{2}$$

In these equations, $\mathrm{T_{symbol}}$ is time taken to transmit each symbol, $\mathrm{N_{payload}}$ is the number of payload bytes , and SF and BW are the spreading factor and bandwidth used.

LoRaWAN is a standardized Medium Access Control (MAC) protocol developed by the LoRa Alliance [18], operating primarily in a star-of-stars topology. It employs an Adaptive Data Rate (ADR) mechanism, allowing the network server to dynamically adjust a node's spreading factor and transmission power based on link quality [19]. LoRaWAN supports three classes of operation, with Class A being the most common for low-power devices. In Class A, each transmission is followed by two receive windows, providing an opportunity for the node to receive an acknowledgment (ACK) for successful communication.

[2]EGAL is derived from "egalitarianism," emphasizing equal energy distribution among all network nodes.

[3]Inactive or non-communicating links are those not used in the current communication but are potential candidates for future communication

[4]For a detailed explanation of the airtime equations and how to calculate $\mathrm{N_{payload}}$, refer [17]

B. Energy Model of EGAL

Each node starts with a fixed battery value and operates in one of three states: Transmission, Reception, and Sleep. The energy consumed in each state is calculated as:

$$\text{Energy}_{\text{state}} = \text{Current}_{\text{state}} \times \text{Time}_{\text{state}} \quad (3)$$

Using SX1272-FiPy node connected to a Monsoon Power Monitor, we measured current consumption for each SF. The final measurements were averaged across all SFs, as variations were negligible, resulting in 150 mA for Transmission (Tx), 60 mA for Reception (Rx), and 20 mA for Sleep. Transmission and reception times are calculated using ToA equations 1 and 2, with the remaining time spent in sleep mode.

Network lifetime is defined as the time at which the first node's battery is completely depleted, taking into account the energy consumption in all states. The battery drain for each node can be expressed as:

$$\text{Battery}_{\text{remaining}} = \text{Battery}_{\text{initial}} - \sum_{\text{state}} (\text{Energy}_{\text{state}}) \quad (4)$$

The network lifetime, therefore, is determined by the *node with the minimum remaining battery.* In LoRa networks, battery disparity arises from different SFs, as each increase in SF doubles the ToA (refer to equation 1). EGAL addresses this by offloading packets from nodes with higher SFs to intermediate nodes with lower SFs, thereby balancing the energy load.

III. Related Work

A. Non-Relay-Based Approach

Various studies [7]–[12] propose advanced algorithms to enhance LoRa's energy efficiency by outperforming standard ADR [19]. These algorithms offer improved allocation of LoRa's physical layer parameters, such as spreading factor, transmission power, and coding rate. However, the issue of energy disparity among nodes operating at different SFs persists. While [20] and [21] address load balancing by assigning a similar number of nodes to each SF, this can sometimes force nodes to use higher SFs than necessary, leading to sub-optimal energy consumption. In EGAL, we mitigate energy disparity while ensuring that each node operates at its optimal SF.

B. Relay-Based Approach

Long-Lived LoRa (LLL) [22] extends network lifetime in energy-harvesting LoRa networks by offloading packets from depleted nodes to those with sufficient energy. We adapted and evaluated LLL in non-energy-harvesting scenarios (see Section VI-A). MLoRa [15] balances energy consumption and extends network lifetime by evaluating relay candidates based on the gain-to-cost ratio. To avoid local optima, MLoRa employs a random Russian roulette method to explore new routes, which can lead to inefficient choices. Similarly, EMH [14] uses an epsilon-greedy reinforcement learning algorithm to enhance LPWAN network lifetime by alternating between exploiting the least energy-consuming path and exploring new ones. EMH initially collects single-hop RSSI information and

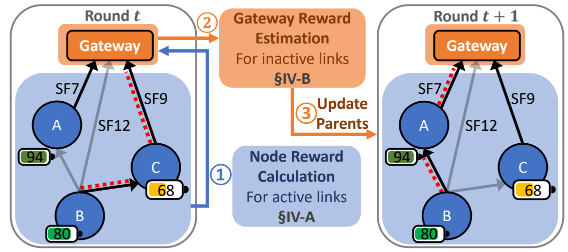


Fig. 2: Overview of the system workflow for EGAL. End-nodes calculate reward values for communicating links and include them in packet transmissions to the gateway. The gateway analyzes these rewards, estimating rewards for non-communicating links, and reassigns parent nodes to balance the network load.

prioritizes links with high RSSI values. However, random exploration can result in sub-optimal decisions, and RSSI values may become stale over time. Other methods [23], [24], [25] improve reliability and energy efficiency by building multi-hop LoRa networks but often introduce significant control overhead. Unlike these approaches, EGAL operates centrally with minimal overhead and avoids exploration costs.

IV. EGAL Algorithm

EGAL is a framework designed to balance energy consumption across end nodes in a LoRa network, thereby extending network lifetime. EGAL employs a hybrid approach, enabling direct communication between end nodes and the gateway at their respective spreading factors (SFs) while also allowing nodes to offload packets to intermediate nodes operating at lower SFs. Node-to-node communications are maintained at a fixed SF, typically SF 7, 8 or 9, based on network deployment specifics. We categorize nodes as *'parent'* or *'child'*: a child node transmits its packets to a parent node, which may be an intermediate relay node or the gateway itself.

In EGAL, we draw inspiration from Reinforcement Learning (RL) [26] approaches, where actions are optimized based on reward calculations to achieve a specific goal. However, conventional RL techniques are not directly applicable in this context for the following reasons: (1) Random exploration of the environment, common in RL, can lead to inefficient energy use in LPWANs, and (2) Changing network conditions can render historical rewards, prior explorations, and actions irrelevant. To address these challenges, EGAL employs an adaptive *reward-based* algorithm that draws from RL principles but is designed specifically to avoid these pitfalls. As shown in Figure 2, EGAL incorporates *Reward Calculation* and *Reward Estimation* based on the current network state to respond to changing conditions and eliminate exploration costs. The gateway, with its centralized perspective, uses these rewards to adjust parent assignments, thereby optimizing energy distribution across nodes.

Figure 3 offers a more granular view of the algorithm. ① Each node calculates the reward associated with its link to a child node upon packet reception (Section IV-A). ②

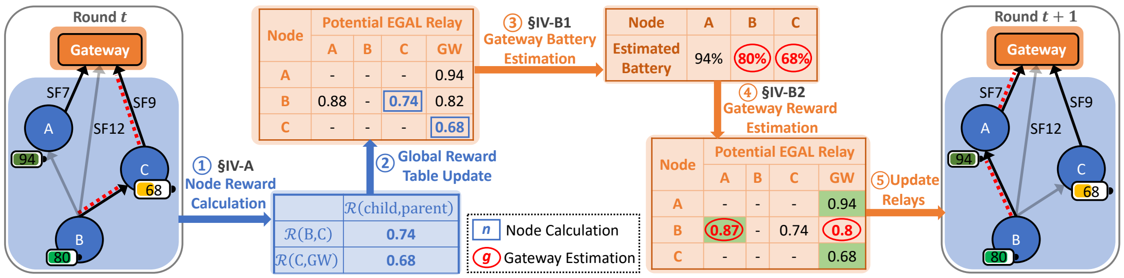


Fig. 3: Detailed overview of EGAL workflow: (1) Nodes calculate reward values based on current network interactions. (2) Nodes update the gateway's *reward table* (3) Gateway estimates the residual battery levels. (4) Rewards for inactive links estimated based on the battery estimations. (5) Gateway reassigns parent nodes with maximum reward (highlighted in green)

The rewards are updated to the *global rewards table* at the gateway. ③ After each communication round[5], the gateway estimates the battery levels of the end nodes (Section IV-B1). ④ The battery levels are used to estimate the reward values of the inactive links that could not update the global reward table (Section IV-B2). Finally, ⑤ the gateway revises the parent node assignments. In the ensuing sections, we provide a detailed discussion of these steps.

A. Reward Calculation by End Nodes

Upon receiving a packet, the parent node P computes the reward for its link with the child node C, denoted $\mathcal{R}(C,P)$:

$$\mathcal{R}(C,P) = \begin{cases} \text{Batt}_C, & P \text{ is Gateway}, \\ \dfrac{\text{Batt}_C + (\text{Batt}_P - E_{\text{queue}})}{2}, & \text{otherwise}. \end{cases} \tag{5}$$

Here, Batt_C and Batt_P are the battery levels of the child and parent nodes after transmission, and E_{queue} estimates the energy needed to transmit queued packets at the parent node P. Including E_{queue} penalizes nodes with large queues, preventing them from being overburdened with relay duties.

The reward value compactly represents the energy efficiency of the link and reducing the need for extensive information transfer to the gateway.

B. Gateway-Based Reward Estimation

The *global reward table* at the gateway provides a centralized view of energy efficiency of potential relay links. The rewards for active links are updated directly by the nodes. However, the changes in the battery levels of these nodes due to communication also affect the rewards for the inactive links. For instance, if node P primarily receives packets from child C_1, it updates $\mathcal{R}(C_1,P)$ actively. However, if P depletes its battery, the rewards for other potential child nodes, $\mathcal{R}(C_i,P)$ need re-calibration.

To ensure that the gateway's decision-making is based on the most recent and relevant data, we introduce *Battery Estimation* for each node, which subsequently enables *Reward Estimation* for the inactive links.

[5]A fixed period during which each node has one opportunity to transmit.

*1) **Battery Estimation**:* At the end of each communication round, the gateway estimates the remaining battery for each node. With knowledge of initial battery levels, data relayed, and the SF used, the gateway can estimate the node's energy expenditure (Equation 3), and therefore the residual battery (Equation 4). This allows for the recalibration of the *stale* rewards values, as discussed next.

*2) **Reward Estimation**:* The gateway uses updated battery estimates to recalibrate stale rewards using Equation 5, assuming negligible E_{queue} since the gateway cannot determine each node's queue size. Node queue size is only considered during actual reward calculations, not in the estimations. Finally, the gateway leverages the updated *global rewards table* to select the most efficient parent P for each child node C as $P = \text{argmax}_i\, \mathcal{R}(C,i)$. The link with the highest reward signifies the most efficient route for packet relay, balancing the battery load across the network.

Reward estimation at the gateway for inactive links eliminates the need for random exploration within the network.

C. Algorithm Flow with Example

Figure 3 illustrates the algorithm with three nodes, A, B, and C, communicating directly with the gateway at SF 7, 9, and 12, respectively. Nodes A and C serve as potential relays or 'parents' for Node B to help preserve battery life.

For simplicity, we assume the communication round comprises a single packet transmitted from Node B to the gateway (GW), routed via Node C. It is assumed that the packet queues are currently empty for all the nodes. The post-transmission battery values are also depicted, with Node C experiencing depletion due to the load it currently faces. The algorithm flow proceeds as follows:

*1) **Node Reward Calculation**:* Node B transmits a packet to Node C, including its battery level. Node C calculates $\mathcal{R}(B,C)$ as 0.74 using Equation 5. With an empty queue assumed, this reward represents the average battery levels of Nodes B and C. For the link between Node C and the gateway, $\mathcal{R}(C,GW)$ is 0.68, reflecting Node C's residual battery.

*2) **Global Reward Table Update**:* Node C relays the data to the gateway with the calculated rewards, which are then updated in the global reward table. More detail on these message exchanges are presented in Section V.

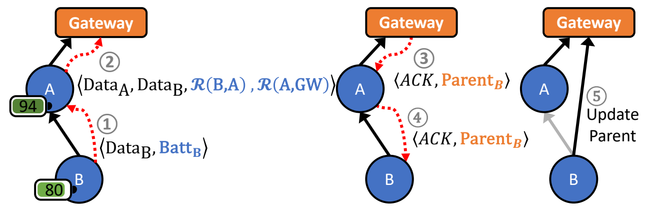


Fig. 4: Passing control information through the network, piggybacking on data packets and acknowledgments (ACKs).

*3) **Gateway Battery Estimation**:* The gateway estimates battery levels for Nodes B and C based on their activity, showing Node C at 68% and Node B at 80% battery. With the knowledge of the total payload the nodes transmit and SFs used, the gateway estimates the residual battery as per Equations 1, 2, 3, and 4. Since Node A was not involved in any communication, its battery level remains unchanged.

*4) **Gateway Reward Estimation**:* No transmission occurred over links $B \leftrightarrow A$ and $B \leftrightarrow GW$, so their rewards could not be updated. However, with a fresh estimate of Node B's battery levels, the gateway re-calibrates $\mathcal{R}(B, A)$ and $\mathcal{R}(B, GW)$ to 0.87 and 0.80, using Equation 5.

*5) **Update Relays**:* Finally, the gateway assigns the relay with the highest reward to each node for the next round $t+1$. Consequently, Node B is assigned Node A as its parent, while Node C continues to communicate directly with the gateway.

This example shows how EGAL adapts to Node C's depleting battery by redirecting its child through an alternative route, reducing the load on Node C.

V. Overall System Design- Integrating EGAL

This section outlines the integration of EGAL into a standard Class A LoRaWAN network, focusing on the modifications made to ensure low overhead.

A. Network Configuration and Setup

Network initialization in EGAL begins with the standard LoRaWAN process, establishing single-hop links and determining each node's one-hop SF to the gateway. This is followed by a neighbor discovery phase for nodes to identify potential parents. While the specifics of the neighbor discovery algorithm are beyond this paper's scope, any standard algorithm can be applied. If node locations are known, neighbor discovery may not be necessary. Given that most LoRa network deployments maintain a static topology, EGAL is well-suited for such static scenarios where frequent neighbor discovery is unnecessary. To reduce complexity, EGAL limits neighbor discovery to a single SF. Although multi-SF discovery could enhance performance by providing more links for load balancing, it would also lead to higher control energy consumption.

In our implementation, the gateway broadcasts the neighbor discovery schedule, after which nodes announce their presence and report discovered neighbors back to the gateway. The gateway then constructs the global reward table based on the identified parent-child combinations. Neighbor discovery is restricted to SF7, causing all inter-node communications to occur at this SF.

B. Communication Phase

*1) **Parent-Child Synchronization**:* EGAL employs time-division multiple access (TDMA) for communication, where all nodes synchronize their time with the gateway during the formation of single-hop links. LoRa's low bit rate reduces the need for frequent synchronization, as the required resolution is coarse (on the order of tens of milliseconds). Each node calculates its wake-up slot based on its assigned ID, ensuring parent and child nodes wake up simultaneously. Time slots repeat after each communication round, adhering to the standard 1% duty cycle regulations, with each node allotted one slot per round.

While EGAL can also operate using an ALOHA-based MAC like LoRaWAN, a time-slot-based MAC is preferred for parent-child synchronization due to reduced overhead. Nonetheless, EGAL is compatible with any MAC protocol.

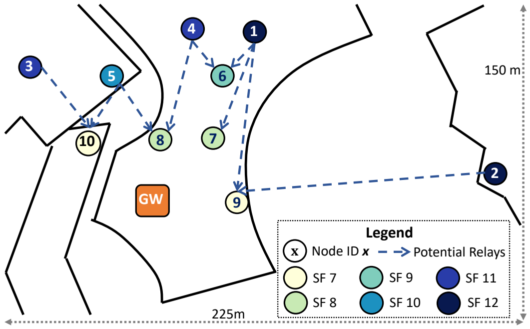
*2) **Data Transmission**:* In EGAL, both node-to-node and node-to-gateway communications utilize message types from the LoRaWAN standard. Node-to-node communication uses the *Proprietary* message type, while node-to-gateway communication employs the *Confirmed Data Up* type, which requires acknowledgment for each uplink frame.

Figure 4 illustrates how EGAL minimizes overhead by piggybacking reward values and parent update decisions onto regular communication packets. Node B transmits its battery level (Batt_B) alongside the payload (Data_B), adding a 4-byte overhead for the battery value. Parent Node A then calculates the reward $\mathcal{R}(B, A)$ and relays Data_B, the rewards $\mathcal{R}(B, A)$ and $\mathcal{R}(A, GW)$, and its own payload (Data_A). Each reward adds 4 bytes of overhead. Consequently, the total overhead can be summarized as $4(n_c + 1)$, where n_c represents the number of child nodes.

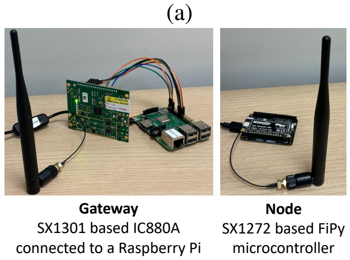
After each transmission, Class A LoRaWAN opens two acknowledgment (ACK) windows. Parent update decisions are included in these ACKs with a 2-byte overhead per update, conveying the child ID and its new parent ID (as shown in steps 3 and 4 of Figure 4). The received updates are implemented in the following round.

In summary, embedding reward values in data packets adds a 4-byte overhead per reward value, while parent updates appended to ACK packets add a 2-byte overhead per update. These control overheads are factored into our energy consumption and battery drain calculations (Equations 3 and 4), allowing for an accurate evaluation of each link's cost.

*3) **Fallback Mechanism**:* If a node fails to reach its intermediate parent after three missed ACKs, it directly transmits to the gateway. The gateway then updates the node with information about a new parent. If the node also fails to reach the gateway at the estimated SF, it resorts to LoRaWAN's fallback mechanism, which allows communication with the gateway at the maximum SF.



(a)



(b)

Fig. 5: a) Deployment Pattern and Topology Formed for Real World LoRa testbed. b) Hardware used in LoRa testbed.

VI. Evaluation Setup

In this section, we present evaluation of EGAL, obtained from both simulations and hardware implementations.

A. Baseline Algorithms

We evaluate EGAL against three baselines: standard LoRaWAN, Multi-Hop LoRa (MLoRa), which combines known information exploitation with random exploration, and Long Lived LoRa (LLL), a heuristic-based method. To ensure fairness, we modified LoRaWAN from its original ALOHA-based MAC to a time slot-based approach, similar to EGAL. This change is essential because the ALOHA-based MAC significantly reduces network lifetime due to increased packet collisions and re-transmissions. By using a consistent link layer across EGAL, LoRaWAN, MLoRa, and LLL, we achieve a fair comparison. For completeness, we also include results for the original LoRaWAN-ALOHA. However, the most accurate reflection of EGAL's performance gains comes from comparing it to LoRaWAN-TDMA.

Analysing LLL in a Non-Energy Harvesting Scenario: LLL enhances network lifetime by offloading packets to intermediate affluent nodes in an energy-harvesting network. A node is considered affluent if its battery level exceeds the amount needed until the next recharge cycle, while a node with less battery is deemed depleted. Depleted nodes are associated with affluent ones until the affluent node's battery drops below the required threshold. This simple greedy association works effectively due to the regular battery recharge cycle.

In a non-energy harvesting scenario, however, there is no battery recharge cycle. Instead, we define an affluent node as one whose battery is above the network's current average battery level. Here, the average battery level acts as a dynamic threshold, analogous to the recharge cycle in the original LLL. Just as the recharge cycle determines which nodes are affluent by providing a predictable point of energy replenishment, the average battery level allows us to identify affluent nodes by comparing them to the network's overall energy status.

The purpose of adpating LLL to a non-energy harvesting network is to assess how a threshold-based greedy approach compares to EGAL. Although our implementation does not exactly replicate LLL, it adapts its core idea. We will refer to this adapted version as **LLL-Modified** in our results.

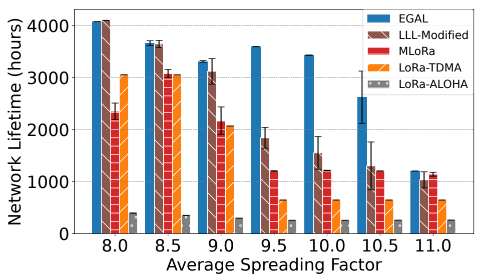
B. Simulator Description

We integrate our algorithm into the existing LoRaWAN module [27] of the NS3 simulator. In our simulation setup, nodes are deployed randomly in a circular configuration with a maximum radius of 5810 meters, and the gateway is positioned at the centre. The duty cycle is set to 1% as per LoRaWAN regulations, with an operating bandwidth of 125KHz. Nodes communicate with the gateway using different spreading factors (SF) ranging from SF7 to SF12. Unless otherwise stated, the direct SF of the nodes to gateway is sampled from a uniform random distribution. Consequently, the average SF of the network ranges between 9 and 10. All node-to-node communication occurs at SF7.

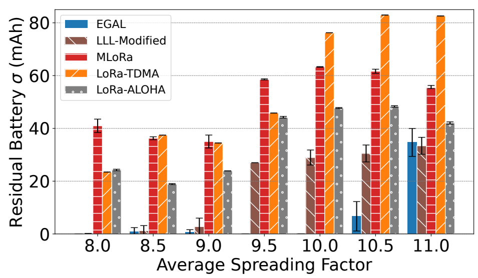
Each node randomly chooses a load based on its operating SF. Lower SFs have higher data rates, and we assign different maximum loads accordingly: SF7 is 1300 bytes, SF8 is 710 bytes, SF9 is 365 bytes, SF10 is 165 bytes, SF11 is 65 bytes, and SF12 is 30 bytes. These maximum loads are calculated based on the length of our communication slots, which are fixed to support one SF12 transmission. The default mean operating load across evaluations is 10% of the maximum load. Instead of running our tests at a fixed load, assigning load to each node based on its operating SF makes the experiments more realistic, as in the real world, lower SFs are used to transmit more data than higher SFs.

C. Configuration of Real-World Testbed

We extend our evaluation to a real-world LoRa testbed, as depicted in the deployment Figure 5a. The number inside the circle of Figure 5a represent node Id's and the colour of the circles indicate their operating SF to the gateway. The indoor testbed spans multiple buildings on campus, covering an area of approximately 33,750 square meters. There are 10 LoRa nodes and 1 gateway deployed in a realistic indoor environment, with various obstacles such as walls, doors, metallic shutters, and crowded common spaces. For the LoRa transmitters, we utilize SX1272-based Fipy microcontrollers. The gateway is equipped with an SX1301-based IC880A, connected to a Raspberry Pi (see Figure 5b). The LoRa network operates with a bandwidth of 125 kHz, and all nodes communicate at a duty cycle of 1%. Similar to the simulator, nodes connect to the gateway at various spreading factors (SF) from SF7 to SF12, while all node-to-node communication



(a)



(b)

Fig. 6: a) Comparison of Network Lifetime Across Network's Average SF Values. b) Standard Deviation of Remaining Node Battery Levels at First Node Depletion.

occurs at SF7. The load distribution across each SF is also consistent with the simulator.

VII. Results

We study the performance of EGAL using simulation while varying the following parameters: topology (Section VII-A), initial battery capacity (Section VII-B), proportion of relay nodes (Section VII-C) and load distribution (Section VII-D). Additionally, we analyse the overhead of EGAL in Section VII-E. Finally, we present the real-world testbed evaluation in Section VII-F.

A. Effect of Topology

We evaluate the impact of varying network topologies on EGAL's performance compared to baseline algorithms. The topologies are adjusted based on the network's average Spreading Factor (SF). A lower average SF indicates that most nodes are closer to the gateway and can use lower SFs for transmission, while a higher average SF suggests that nodes are farther away. Each node starts with a battery level of 250 mAh, and results are averaged across three trials, each with random node placements.

*1) **Network Lifetime**:* Figure 6a illustrates that EGAL consistently delivers the best network lifetime across all topologies. The most significant improvement is observed when the average SF ranges from 9.5 to 10.5, where EGAL maximizes relay opportunities to enhance network lifetime. On the left side of the graph, where lower SF nodes dominate, packets reach the gateway at a lower energy cost, reducing the need for relaying. On the right side, where higher SF nodes dominate, all algorithms struggle due to the limited availability of parent nodes for relaying packets.

EGAL's performance aligns closely with threshold-based solutions like LLL when distributions are skewed towards lower SFs. In these cases, both EGAL and LLL identify most nodes as having sufficient energy, resulting in minimal relaying. However, when there are abundant opportunities to relay and extend network lifetime, EGAL outperforms all baselines. Specifically, at an average SF of 9.5, EGAL extends the network lifetime by 199% over MLoRa, 95% over LLL-Modified, and 457% over standard LoRaWAN-TDMA. In contrast, MLoRa consistently underperforms, even dropping below LoRaWAN-TDMA at an average SF of 8, due to the high overhead associated with its random exploration strategy.

*2) **Residual Battery**:* We also examine the standard deviation (σ) of residual battery levels across the network when the first node depletes its battery. A high standard deviation indicates significant imbalance, while a low standard deviation suggests well-balanced battery usage. As shown in Figure 6b, EGAL maintains a substantially lower standard deviation, demonstrating effective battery load balancing. Some imbalance is observed only at an average SF of 11, where relay opportunities are minimal, while the imbalance remains almost negligible until an average network SF of 10.

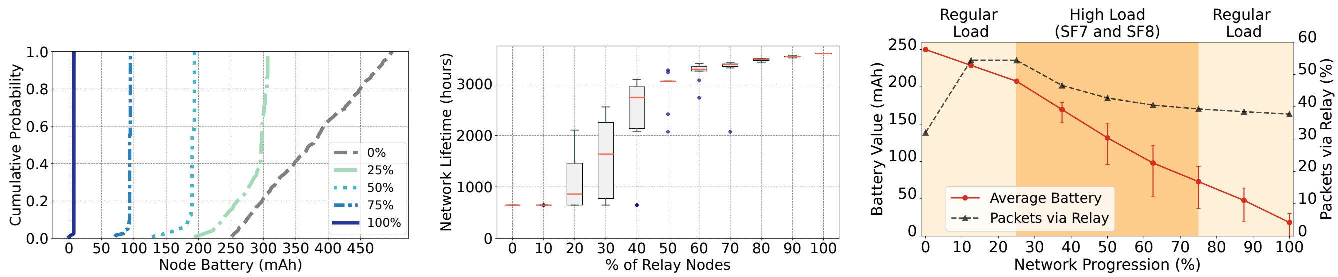
While LLL also demonstrates low battery variance at lower SF distributions, its effectiveness diminishes from an average SF of 9.5, when there is opportunity for relaying. On the other hand, MLoRa and standard LoRaWAN consistently exhibit high battery variance across all topologies.

It's important to note that LoRaWAN-ALOHA consistently underperforms due to frequent packet collisions in large networks, leading to numerous retransmissions. The average Packet Reception Rate (PRR) for LoRaWAN-ALOHA across all SF distributions is approximately 70%, while the other algorithms, utilizing TDMA-based link layers, achieve nearly 100% PRR. Therefore, the performance of EGAL is best compared against LoRaWAN-TDMA. *From this point forward, any reference to LoRaWAN indicates TDMA-based LoRaWAN.*

B. Different Initial Battery Levels

In this section, we analyze the behavior of EGAL in a network where nodes have different initial battery capacities. Each node's initial battery value is randomly selected from a uniform distribution ranging between 250 mAh and 500 mAh. The results are presented in Figure 7a, where the x-axis represents the battery range, while the y-axis shows the cumulative probability of nodes within that range. We plot the cumulative distribution function (CDF) of battery levels from the initial phase (0%) to the final phase, when the first node depletes its battery (100%). Five CDFs are plotted, each representing a different stage of the network's progression.

As shown in Figure 7a, the range of battery levels narrows as the network progresses. Initially, there is a 250 mAh disparity between the nodes with the lowest and highest battery levels. However, EGAL effectively balances the load across all nodes,



(a) (b) (c)

Fig. 7: a) CDF of Node Battery Levels over the lifetime of EGAL Network, b) EGAL's Performance with Varying Number of Relays, c) Analysis of Battery Disparity and Packet Relay Rates Across Run-Time Load Variations

reducing this gap to approximately 50 mAh at the mid-point of the experiment. By the conclusion, nearly all nodes have depleted their batteries simultaneously. This evaluation demonstrates EGAL's strong capability in balancing network's energy consumption.

C. Varying Percentage of Relay Nodes

In this experiment, we investigate how varying the percentage of nodes allowed to provide relay support affects EGAL's performance. We tested scenarios with no relay nodes (0%) and gradually increasing up to a scenario where all potential relays were active (100%). Each configuration was tested across 10 trials, with different groups of nodes randomly selected to provide relay support in each trial.

The results, illustrated in Figure 7b, reveal that allowing just 30% of nodes to act as relays increases the average network lifetime by approximately 2.46x compared to the no-relay scenario. Even with only 20% of nodes acting as relays, EGAL achieves a significant improvement, with the upper limit reaching 60% of the network lifetime observed in the full relay (100%) scenario. This suggests that network designers can strategically plan the deployment to designate a small subset of nodes as relays to ensure that nodes with higher SFs have sufficient options for relaying packets.

Additionally, as the percentage of nodes providing relay support increases, the variance in network lifetime across the trials decreases. This reduction becomes particularly significant when 50% or more of the potential relays are active, leading to more consistent network performance.

In summary, *not all lower SF nodes need to provide relay support.* The results indicate that even with a fraction of nodes available as relays, EGAL significantly extends the network lifetime.

D. Run-Time Load Variation

In this section, we examine how EGAL adapts to run-time changes in network load. Initially, all nodes operate under their regular operating load (as detailed in Section VI-B) and start with a battery level of 250 mAh. At 25% of the network's lifetime, the load on nodes operating at direct SF 7 and 8 is tripled. This high-load phase continues until 75% of the network's lifetime, after which the nodes return to their regular load for the remaining time. This increase in load specifically targets SF7 and SF8 nodes because these act as relays for the majority of high-SF nodes, making this a particularly challenging scenario for load balancing.

Figure 7c presents the results, with error bars representing the variance in remaining battery capacity among the nodes. In the initial phase, EGAL effectively maintains low battery variance, indicating balanced energy consumption across the network.

However, once the high-load phase begins, battery disparity increases as nodes operating at SF7 and SF8 expend more energy to manage their increased load. To balance energy consumption more effectively, EGAL reduces the overall number of packets relayed via these nodes, which helps lower the battery disparity by the end of the high-load phase.

When the network load returns to normal, EGAL continues to reduce battery variance. However, the number of packets sent through relay nodes does not immediately return to the levels seen in the initial phase because the SF7 and SF8 nodes have expended significant energy during the high-load phase. This adaptability demonstrates EGAL's ability to maintain balanced energy usage under varying network conditions.

E. Overhead Analysis of EGAL

We conducted an overhead analysis of the EGAL algorithm by simulating a network with 120 nodes, where one-hop SFs to the gateway were uniformly and randomly distributed between SF7 and SF12. This configuration maximized the number of potential relay nodes, allowing us to assess the maximum possible overhead when EGAL fully leverages relay opportunities. The algorithm dynamically selected paths to the gateway, with options ranging from a direct 1-hop route to up to 6 hops (e.g., SF12 relaying through SF11 down to SF7). The resulting average hop count was approximately 2, indicating a strategic balance between direct gateway access and multi-hop relays to optimize energy usage. Individual packet overheads ranged from 0.48% to 18.1%, with an average control overhead of 2.9% per packet, ultimately extending network lifetime by 94% compared to standard LoRaWAN.

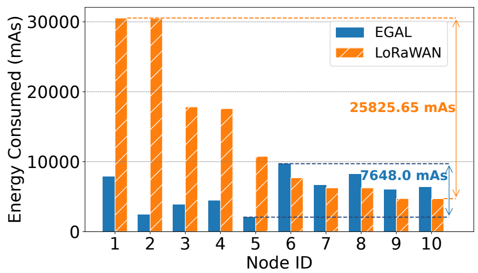


Fig. 8: Real-world Deployment Results: Energy Consumption by Nodes in EGAL and LoRaWAN Networks

F. Real-World Deployment

Figure 8 presents the results comparing EGAL and LoRaWAN in a real-world deployment. We ran the testbed for 24 hours and measured the total energy consumed by each node at the end of this period. The results indicate that LoRaWAN exhibits significant energy disparity, with a difference of 25,826 mAs between nodes 1-2 (operating at SF-12) and nodes 9-10 (operating at SF-7). In contrast, EGAL effectively balances battery consumption, reducing the disparity significantly to 7,648 mAs. Notably, the network topology (Figure 5a) is imbalanced, as not all nodes have an equal number of potential relays. Despite this, EGAL significantly improves energy balance compared to LoRaWAN.

VIII. Conclusion

We introduced EGAL, a relay-based centralized approach that balances energy consumption and extends network lifetime in LoRa networks. EGAL's ability to dynamically adjust relay duties, supported by predictive analytics at the gateway, makes it a robust solution for improving the longevity of IoT networks. Through both simulations and real-world testbed evaluations, EGAL demonstrated substantial improvements over standard LoRaWAN and existing solutions, effectively addressing battery imbalance and enhancing network lifetime with minimal overhead. Furthermore, EGAL's reward-based framework offers promising potential to meet diverse network objectives by expanding the reward function to include metrics such as link quality, latency, and battery discharge rate.

Acknowledgement

We thank the reviewers for their invaluable feedback. This research is supported by National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme.

References

[1] L. Alliance, "A technical overview of lora and lorawan," *White Paper, November*, vol. 20, 2015.

[2] U. Noreen, A. Bounceur, and L. Clavier, "A study of lora low power and wide area network technology," in *International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2017.

[3] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, 2015.

[4] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, "Review of iot applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, 2017.

[5] M. T. Lazarescu, "Design of a wsn platform for long-term environmental monitoring for iot applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.

[6] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the internet of things (iot), applications, opportunities, and challenges: A survey," *IEEE Access*, vol. 8, pp. 69200–69211, 2020.

[7] W. Gao, Z. Zhao, and G. Min, "Adaplora: Resource adaptation for maximizing network lifetime in lora networks," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pp. 1–11, 2020.

[8] Y. Li, J. Yang, and J. Wang, "Dylora: Towards energy efficient dynamic lora transmission control," in *IEEE INFOCOM 2020*.

[9] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in lora networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 788–798, 2019.

[10] B. Teymuri, R. Serati, N. A. Anagnostopoulos, and M. Rasti, "Lp-mab: Improving the energy efficiency of lorawan using a reinforcement-learning-based adaptive configuration algorithm," *Sensors*, vol. 23, 2023.

[11] G. Huang, P. Yang, H. Zhou, Y. Yan, X. He, and X. Li, "Freeback: Blind and distributed rate adaptation in lora-based backscatter networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021.

[12] B. Su, Z. Qin, and Q. Ni, "Energy efficient uplink transmissions in lora networks," *IEEE Transactions on Communications*, vol. 68, 2020.

[13] S. Barrachina-Muñoz, B. Bellalta, T. Adame, and A. Bel, "Multi-hop communication in the uplink for lpwans," *Computer Networks*, vol. 123.

[14] S. Barrachina-Muñoz, T. Adame, A. Bel, and B. Bellalta, "Towards energy efficient lpwans through learning-based multi-hop routing," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019.

[15] S. Feng, J. Chen, and Z. Zhao, "Cost effective routing in large-scale multi-hop lora networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022.

[16] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM Trans. Sen. Netw.*, vol. 15, feb 2019.

[17] P. S. Cheong, J. Bergs, C. Hawinkel, and J. Famaey, "Comparison of lorawan classes and their power consumption," in *2017 IEEE Symposium on Communications and Vehicular Technology (SCVT)*, pp. 1–6, 2017.

[18] M. A. M. Almuhaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on lorawan technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, 2022.

[19] J. Finnegan, R. Farrell, and S. Brown, "Analysis and enhancement of the lorawan adaptive data rate scheme," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7171–7180, 2020.

[20] M. Hamnache, R. Kacimi, and A.-L. Beylot, "Joint load-balancing and power control strategy to maximize the data extraction rate of lorawan networks," *Comput. Netw.*, vol. 225, apr 2023.

[21] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "Explora: Extending the performance of lora by suitable spreading factor allocations," in *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2017.

[22] S. Fahmida, V. P. Modekurthy, M. Rahman, A. Saifullah, and M. Brocanelli, "Long-lived LoRa: Prolonging the lifetime of a LoRa network," in *IEEE International Conference on Network Protocols (ICNP)*, 2020.

[23] M. S. Aslam, A. Khan, A. Atif, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "Exploring multi-hop lora for green smart cities," *IEEE Network*, vol. 34, no. 2, pp. 225–231, 2020.

[24] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, "Enabling rpl multihop communications based on lora," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, 2017.

[25] J. J. L. L. Escobar, F. J. Gil-Castiñeira, and R. P. D. Redondo, "Jmac protocol: A cross-layer multi-hop protocol for lora," *Sensors*, vol. 20, 2020.

[26] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

[27] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of lorawan performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, 2020.