Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Energy-Efficient Routing Protocols in IoT Networks: A Narrative Review and Comparative Simulation Study

Mohammad Haribat

m.haribat@aol.com

Sohar University

Ahmad Kayed

Sohar University

Research Article

Keywords: Internet of Things, Wireless sensor networks, Energy-efficient routing, Network lifetime, Clustering, RPL, Reinforcement learning, Simulation

Posted Date: May 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9578813/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

The battery life of IoT sensors is often the limiting factor, and the choice of routing protocol directly affects how quickly nodes deplete their power. Although many protocols have been proposed, published results use disparate configurations and cannot be compared directly. This paper accomplishes two objectives. First, we give a narrative review of six protocol families: hierarchical, flat, location-based, multipath, QoS- or standard-based, and learning-based. Second, we use the first-order radio energy model of Heinzelman et al. to build an open-source Python/NumPy simulator and run a side-by-side comparison of six reference protocols: FLAT, LEACH, EWC (inspired by HEED), PEGASIS, SSPT (inspired by RPL), and QL-F (a minimal baseline Q-learning implementation included for comparison only). In a 100 × 100 m field with 100 nodes and a central sink, QL-F achieves the longest first-node-death time (1411.1 ± 45.1 rounds), followed by FLAT (1333.9 ± 47.3). A dedicated sink-placement study shows that, across the five positions tested, FLAT loses 65% of its lifetime when the sink moves to a corner, whereas QL-F's first-node-death time stays within 5% of its central-sink value — a robustness aspect that standard evaluations overlook. The simulator is released under an open-source licence to support reproducibility.

Introduction

The Internet of Things (IoT) has evolved over the past decade from a research concept to a widely adopted technology in industry. Smart cities, precision agriculture, industrial monitoring, connected healthcare, and environmental sensing all rely on large numbers of low-power wireless sensors [1,2]. Two recent examples make the point. Medeiros et al. [26] tested several routing protocols on LoRa mesh networks for smart-city sensing, and Yao et al. [27] designed a clustering protocol for precision-agriculture deployments. Both papers show that the choice of routing protocol has a direct effect on how long the network lasts. Most of these sensors share the same constraints: they run on small batteries, have limited memory and processing power, and communicate over radio links that are noisy and low-bandwidth by design [3].

For these devices, radio communication is where most of the energy is consumed. Measurements on low-power platforms such as TelosB show that the radio dominates the per-bit energy budget, costing one or two orders of magnitude more than executing the same computational task on the same chip [4]. Routing protocols are important because they determine how far and how often a node transmits data, which together account for most of the communication energy [5,6].

A practitioner attempting to select a routing protocol faces two immediate challenges. First, the results in the literature cannot be compared directly because they are based on very different assumptions — the radio model, traffic pattern, field size, and number of nodes. Second, the recent rise of learning-based routing has introduced a new class of protocols that are rarely tested on the same setup as the classical baselines they aim to outperform.

Our goal here is narrow and practical. We are not proposing a new protocol; QL-F is a minimal baseline reinforcement-learning implementation included for comparison purposes only. Instead, we provide a

small, easy-to-read simulator and use it to run six reference implementations under identical inputs. The six cover the main design paradigms: direct-to-sink, cluster-based (two variants), chain-based, tree-based, and learning-based. The aim is to offer other researchers a clear baseline that they can use, modify, or disagree with. Our contributions are:

- A narrative six-family taxonomy of energy-efficient IoT routing protocols, focused on design principles rather than exhaustive listing.
- An open-source Python/NumPy simulator that implements six reference protocols under a common first-order radio energy model.
- A quantitative comparison of first-node-death time, per-packet energy, and sensitivity to network size, each averaged over multiple independent Monte Carlo runs (5–30 per experiment, see Section 5).
- A sink-placement sensitivity study across five positions that reveals a 65% lifetime gap between protocols under realistic (non-central) sink geometries — a robustness property that single-configuration benchmarks miss.
- An explicit, honest discussion of what this comparison does and does not show, with concrete directions for future work.

The remainder of the paper is organised as follows. Section 2 presents background on IoT architectures and the energy model. Section 3 describes the review methodology. Section 4 presents the six-family taxonomy. Section 5 describes the simulator and evaluation setup. Section 6 reports the simulation results, including the sink-placement sensitivity study in Section 6.3. Section 7 discusses limitations and open research directions. Section 8 concludes.

Background and Preliminaries

2.1 IoT Network Architecture

Most IoT deployments follow a three-layer pattern: sensors at the edge, a network layer that delivers their data to a gateway, and applications on top that consume it [8]. Routing lives in the network layer, but it cannot ignore what the MAC and physical layers are doing — the three together determine how much energy each bit actually costs. Formally, we model the network as a directed graph G = (V, E), where V represents the nodes and E denotes the feasible wireless links. Each node v is characterised by a triple (E_v, P_v, L_v): residual energy, transmit power, and position.

2.2 Sources of Energy Consumption

Energy dissipation in an IoT node can be decomposed into four components: radio transmission (E_TX), radio reception (E_RX), idle listening and overhearing (E_idle), and computation/sensing (E_comp). Multiple empirical studies report that E_TX + E_RX + E_idle account for 65–85% of total energy drain in

duty-cycled sensor nodes [4, 9]. Routing protocols that reduce the number of transmissions, the per-transmission distance, or the time spent idle-listening therefore yield the largest energy savings.

2.3 First-Order Radio Energy Model

Throughout this paper, we adopt the first-order radio model of Heinzelman et al. [7]. The energy required to transmit an l-bit packet across distance d is given by:

$$E_{TX}(l, d) = l \cdot E_{elec} + l \cdot \varepsilon_{fs} \cdot d^2, \; d < d_0 \quad (1)$$

$$E_{TX}(l, d) = l \cdot E_{elec} + l \cdot \varepsilon_{mp} \cdot d^4, \; d \geq d_0 \quad (2)$$

$$E_{RX}(l) = l \cdot E_{elec} \quad (3)$$

where E_elec is the per-bit electronic energy (typically 50 nJ/bit), ε_fs is the free-space amplifier coefficient (10 pJ/bit/m²), ε_mp is the multipath-fading coefficient (0.0013 pJ/bit/m⁴), and $d_0 = \sqrt{(\varepsilon_{fs} / \varepsilon_{mp})} \approx 87$ m is the distance threshold separating the two propagation regimes. The cost of aggregation is modelled as an additive term E_DA = 5 nJ/bit per aggregated signal. This model is used in almost every paper on energy-efficient routing, so our numbers are consistent with what others have reported. Its main weakness — worth stating up front — is that it does not account for shadowing, fading, or MAC-layer contention. We return to this point in Section 7.

2.4 Network Lifetime Definitions

Three lifetime metrics appear repeatedly in this literature. FND (First-Node-Death) is the round at which the first node runs out of battery. HND (Half-Node-Death) is when half the nodes have died. LND (Last-Node-Death) is when no node remains alive. FND is the strictest of the three and is the metric we use as our main figure of merit. In most deployments every node matters, and any protocol that improves FND usually improves HND and LND as well.

Review and Simulation Methodology

3.1 Literature Review Approach

The review portion of this paper is a narrative review rather than a systematic one. We did not apply the PRISMA protocol or the formal SLR guidelines [25], and we did not use quantitative inclusion criteria. Instead, we surveyed representative papers from each of the six protocol families over the 2002–2025 period, with a focus on the foundational works that define each family and on recent surveys [10–14] that cover current developments. The aim is to place the simulation study in context, not to provide an exhaustive bibliography. For readers seeking a fully systematic treatment, we recommend the recent comprehensive surveys [11, 13].

3.2 Research Questions

The paper is guided by three questions:

- RQ1. What are the main design paradigms used by energy-efficient routing protocols for IoT, and what trade-offs do they embody?
- RQ2. Under a single shared simulation setup, how do representative protocols from each paradigm compare in terms of first-node-death, per-packet energy, and sensitivity to network size?
- RQ3. What are the most important limitations of such unified comparisons, and what directions would strengthen future evaluations?

3.3 Simulation Methodology

To answer RQ2, we used a small simulator written in Python 3 and NumPy. It runs in discrete rounds: in each round, every alive node generates one packet of l bits, and the protocol decides how to deliver that packet to the sink. Per-packet energy is accumulated using Eqs. (1)–(3). The number of independent Monte Carlo runs is set per experiment to balance statistical confidence and computational cost: 30 runs for the main lifetime comparison at N = 100 (Section 6.1), 10 runs for the network-size sweep across $N \in \{50, 100, 200\}$ (Section 6.2), and 5 runs for the five-position sink-placement study (Section 6.3). For each run we use a different random seed, and we report the mean and standard deviation. The code is openly available — see the Data Availability statement at the end of the paper.

Taxonomy of Energy-Efficient Routing Protocols

The surveyed protocols fall naturally into six families, summarised in Table 1. The families overlap — recent protocols often combine two or more mechanisms — but we classify each by its dominant idea.

Table 1
Six-family taxonomy of energy-efficient IoT routing protocols.

Family	Core Mechanism	Representative Protocols
Hierarchical (cluster-based)	Nodes self-organise into clusters; cluster heads aggregate and forward data to sink	LEACH [7], HEED [15], TEEN [16], SEP [17], DEEC [18]
Flat	All nodes play an equivalent role; data-centric or gradient-based forwarding	Directed Diffusion, SPIN, Rumor Routing
Location-based	Uses geographic coordinates to forward toward sink	GEAR [19], GAF, PEGASIS [20]
Multipath	Maintains multiple paths for reliability and load balancing	EAMR, N-to-1 Multipath
QoS / standard-based	IETF-standardised or QoS-aware routing for constrained IP networks	RPL [21], CORPL, ORPL, LOADng
Learning-based	Uses reinforcement learning or metaheuristics to adapt routes to observed conditions	Q-Routing, ACO-Routing, DRL-RPL [22, 23]

4.1 Hierarchical (Cluster-Based) Routing

This is the largest family in the literature. The best-known member is LEACH [7], which rotates the cluster-head role between nodes using a random threshold $T(n) = p/(1 - p\cdot(r \bmod 1/p))$, where p is simply the target fraction of cluster heads. Since LEACH came out, many papers have built on it. HEED [15] selects cluster heads based on residual energy rather than random chance. SEP [17] and DEEC [18] target heterogeneous networks in which nodes start with different amounts of energy. TEEN [16] only transmits data when the sensed value crosses a threshold, which sharply reduces the number of transmissions when the signal is stable. Cluster-head selection is still an active research topic. Yao et al. [27] borrow ideas from multi-threshold image segmentation to handle both cluster formation and head election, and report up to 64% less energy consumption than RLEACH on precision-agriculture topologies.

The weaknesses of this family are well documented. When the sink is located at one edge of the field, nodes near it deplete their energy first — the classic energy-hole problem. Cluster re-formation also incurs control-message overhead every round, and that cost accumulates over thousands of rounds. Moreover, when a cluster head is far from the sink, the radio model switches to the expensive d^4 regime and consumes a large amount of energy on a single hop.

4.2 Flat Routing

Flat protocols abandon the role hierarchy and use data-centric primitives. Directed Diffusion and SPIN are canonical examples. Although conceptually simple, flat routing generally scales poorly beyond a few hundred nodes because of flooding overhead, and it has declined in the post-2020 literature.

4.3 Location-Based Routing

Location-based protocols leverage geographic information (GPS or localisation algorithms) to forward greedily toward the sink. GEAR [19] combines greedy geographic forwarding with residual-energy weighting. PEGASIS [20] organises nodes in a chain traversed once per round, achieving good per-round energy but incurring long latency for the farthest node.

4.4 Multipath Routing

Multipath protocols keep several paths from source to sink instead of just one. The paths may be fully node-disjoint or share some nodes. Energy-Aware Multipath Routing switches among them based on residual energy, which spreads the load and extends network lifetime. The price is more memory and more signalling traffic.

4.5 QoS and Standard-Based Routing

RPL [21] is the heavyweight in this family. It was standardised by the IETF in RFC 6550 back in 2012. The idea is straightforward in principle: the sink roots a Destination-Oriented Directed Acyclic Graph (DODAG), and every other node selects a parent based on an Objective Function. Two OFs are part of the standard. OF0 uses hop count, while MRHOF uses a link-quality metric (usually ETX).

Researchers have proposed many RPL variants — ORPL adds opportunistic forwarding, and CORPL handles cognitive-radio scenarios — but none of them has been standardised. RPL has one practical advantage worth stating clearly: it is the only IoT routing protocol that ships out-of-the-box in Contiki-NG, OpenWSN, RIOT OS, and Zephyr. That is the reason most new proposals end up comparing against it.

4.6 Learning-Based Routing

This is the fastest-growing family over the last three years [12, 13]. Q-routing casts forwarding as a Markov Decision Process: the state S captures node identity and the residual energies of all nodes, the action A is the choice of next-hop neighbour, and the reward combines energy and delay. Recent extensions integrate residual-energy and hop-count signals into the reward function to balance lifetime and end-to-end delay [22]. A related sub-family uses swarm intelligence — ACO-Routing borrows the pheromone update idea from ant-colony optimisation. Newer metaheuristic-based work goes further: Darabkh and Al-Akhras [28] use the Marine Predators Algorithm to pick cluster heads and combine that with data fusion to cut down on duplicate transmissions, which extends lifetime even more. Three caveats to keep in mind: the exploration phase itself costs energy, formal convergence guarantees are hard to prove in mobile settings, and the memory footprint can be a problem on Class-0 or Class-1 devices.

Simulator and Evaluation Setup

5.1 Protocols Implemented

This study presents a paradigm-level comparison using simplified reference implementations; results are not intended to represent exact protocol performance. To be transparent about what we actually ran, we give each reference implementation a short name that distinguishes it from the published protocol that inspired it. The six implementations omit the engineering details (MAC integration, control-plane messaging, timer tuning) that the original protocols rely on, but they capture the core forwarding logic of each design paradigm under the same energy model — which is what we need for a like-for-like comparison at the paradigm level.

- FLAT — every alive node transmits its packet directly to the sink in every round. No coordination, no aggregation.
- LEACH — cluster-head rotation following the stochastic threshold rule of Heinzelman et al. [7] with p = 0.05. Implemented as described in the original paper.
- EWC (Energy-Weighted Clustering) — a clustering variant inspired by HEED [15] in which the cluster-head election probability is scaled by the node's residual energy. It is not a faithful HEED implementation; the iterative probability-doubling and tie-breaking rules of HEED are not modelled.
- PEGASIS — a greedy nearest-neighbour chain built once per experiment with the leader role rotating every round, as in Lindsey and Raghavendra [20].

- SSPT (Static Shortest-Path Tree) — a tree rooted at the sink, in which each node's parent is the in-range neighbour closest to the sink. This is inspired by RPL [21] but is not RPL: we do not model DIO/DAO messaging, trickle timers, or dynamic parent selection. SSPT is a static routing tree, intended as a lightweight stand-in for tree-based routing.
- QL-F (Q-learning Forwarding) — a minimal baseline reinforcement-learning implementation, included for comparison purposes only and not intended as a novel protocol contribution. Each node maintains Q-values for its feasible next-hop candidates (in-range neighbours closer to the sink, plus a direct-to-sink action) and uses an ε-greedy policy with ε decaying linearly from 0.15 to 0.02 over the simulation horizon. QL-F is not drawn from any published protocol; it is a simple reinforcement-learning representative for the comparison.

We want to be explicit on one point: EWC, SSPT, and QL-F are reference stand-ins for their respective paradigms (energy-weighted clustering, tree-based routing, learning-based routing). They are deliberately simple so that the comparison is transparent and reproducible. A reader interested in the performance of HEED, RPL, or a published Q-learning router specifically should consult the original papers or implement those protocols in NS-3 or Cooja — we discuss this in Section 7.

5.2 Simulation Parameters

Table 2 lists the simulation parameters. All protocols share the same random topology per seed; the number of independent seeds depends on the experiment, as detailed in Section 5.1.

Table 2
Simulation parameters.

Parameter	Symbol	Value
Field side length	L	100 m
Sink position	—	(50, 50) m
Number of nodes	N	50, 100, 200
Initial energy per node	E_0	0.5 J
Packet length	l	4000 bits
Electronic energy	E_elec	50 nJ/bit
Free-space amplifier coefficient	ε_fs	10 pJ/bit/m²
Multipath amplifier coefficient	ε_mp	0.0013 pJ/bit/m⁴
Aggregation energy	E_DA	5 nJ/bit/signal
Transmission range (SSPT, QL-F)	R_tx	30 m
Maximum rounds	R_max	2000
Independent runs	—	30 (Sec. 6.1) / 10 (Sec. 6.2) / 5 (Sec. 6.3)

5.3 Evaluation Metrics

We report three numbers. The first is FND (First-Node-Death time): the round count until the first node in the network dies. The second is E_pkt, the average energy per delivered packet — we take the total energy burned by the network and divide it by the total number of packets that actually reached the sink. The third is how E_pkt changes with network size, measured at N = 50, 100, and 200.

Simulation Results

6.1 Network Lifetime (First-Node-Death)

Table 3 gives the FND, HND, and LND of the six implementations at N = 100, averaged over 30 runs. QL-F achieved the longest FND at 1411.1 ± 45.1 rounds. FLAT came in second at 1333.9 ± 47.3, and EWC was third at 947.9 ± 24.2. At the other end of the table, LEACH had the shortest FND (624.1 rounds). The cause is straightforward: LEACH's random cluster-head rotation occasionally picks a low-energy node close to the sink, that node dies fast, and from there the whole network starts to unravel. SSPT did not fare much better (695.6). Its problem is different but the outcome is similar — the static tree puts a few relay nodes near the sink in charge of forwarding traffic for many descendants, and those relays run out of battery early. A real RPL implementation would periodically rebuild the tree around residual energy using an energy-aware objective function, and SSPT would likely perform better as a result; we did not implement that.

Table 3
Lifetime and per-packet energy at N = 100 (mean ± std across 30 runs, 2000 rounds maximum).

Protocol	FND (rounds)	HND	LND	E_pkt (mJ)
LEACH	624.1 ± 54.3	998.0	1506.9	9.43
SSPT	695.6 ± 85.9	1526.8	2000	0.97
PEGASIS	826.5 ± 177.2	1176.7	1695.8	29.59
EWC	947.9 ± 24.2	1048.5	1067.5	17.19
FLAT	1333.9 ± 47.3	1892.4	2000	0.26
QL-F	**1411.1 ± 45.1**	1805.2	2000	0.31

The FLAT result deserves some explanation because it pushes back against the usual intuition that direct-to-sink is always the worst choice. In our setup the sink sits at the centre of a 100 × 100 m field, so the average source-to-sink distance is around 36 m — comfortably below the threshold $d_0 \approx 87$ m at which the radio model shifts from the cheap d^2 free-space regime to the much more expensive d^4 regime. Keep the sink central and the field small, and FLAT stays in the cheap regime for every transmission. Move the sink to a corner, or push L past 175 m or so, and that stops being true. Section

6.3 runs this experiment explicitly and the drop is substantial. Figure 1 visualises the FND ranking for all six protocols.

6.2 Per-Packet Energy vs. Network Size

Table 4 shows how per-packet energy responds to the number of nodes, averaged over 10 runs of 1500 rounds at each size. The protocols fall into two clean groups. FLAT, SSPT, and QL-F deliver one packet per alive node per round, and their per-packet energy stays low and roughly constant. LEACH, EWC, and PEGASIS deliver far fewer packets per round — one per cluster head in LEACH and EWC, one per round for the whole chain in PEGASIS — so they end up amortising a lot of intra-cluster or intra-chain transmissions over a small number of deliveries. PEGASIS is the extreme case: doubling N from 100 to 200 almost doubles E_pkt, because the chain is now twice as long but still produces a single packet per round. Whether this matters depends on the application. If raw energy per delivered byte is what you care about, PEGASIS is expensive. If what you care about is how long the network stays useful, PEGASIS does less badly than this table suggests, because it keeps more nodes alive for longer.

Table 4
Per-packet energy (mJ) vs. network size N (mean across 10 runs, 1500 rounds).

Protocol	N = 50	N = 100	N = 200
FLAT	0.266	0.265	0.266
QL-F	0.300	0.310	0.328
SSPT	1.080	1.036	1.002
LEACH	9.106	9.476	9.780
EWC	15.261	17.074	16.770
PEGASIS	16.652	33.374	67.562

6.3 Sink-Placement Sensitivity

The Section 6.1 FLAT result made us suspicious. A protocol that wins on lifetime just because the sink happens to sit in the middle is not really winning — it is benefiting from the geometry. To see how brittle that advantage is, we ran the six protocols at five sink positions in the same 100 × 100 m field: Center (50, 50), Near-Edge (50, 30), Edge (50, 10), Corner (10, 10), and Far-Corner (5, 5). Each configuration was run 5 times with different random seeds, at N = 100, for up to 1000 rounds. Figure 3 shows the first-node-death time for each protocol across the five sink positions.

The picture is striking. FLAT drops from 1000 rounds at the centre to about 347 at the far corner — a 65% loss. The reason is the energy model: with a corner sink, the farthest nodes are now more than 120 m away, which is well into the d^4 multipath regime, and every packet those nodes send costs an order of magnitude more energy than the central case. The cluster-based protocols (LEACH, EWC) and PEGASIS

barely move, because their cluster heads or chain aggregate most transmissions into short hops. QL-F is the most interesting result here: it loses less than 5% of its lifetime, because the ε-greedy policy quickly learns to route through intermediate neighbours rather than pay the d^4 cost of a direct transmission. SSPT sits at the bottom of the chart throughout, which reflects its static tree — the nodes near the sink are always the same overloaded relays, regardless of where the sink is. Table 6 gives the raw numbers.

Table 6
FND (rounds) across five sink positions (mean of 5 runs, N = 100, 1000 rounds maximum).

Protocol	Center	Near-Edge	Edge	Corner	Far-Corner
FLAT	1000	1000	753	423	347
LEACH	1000	1000	1000	1000	1000
EWC	1000	1000	1000	1000	1000
PEGASIS	1000	1000	1000	1000	1000
SSPT	200	256	267	275	275
QL-F	**1000**	**1000**	**1000**	**985**	**945**

The practical takeaway is that rankings based on a single sink position can mislead. In a realistic deployment the sink is rarely at the geometric centre — gateways are often placed at a building edge, a pole, or a road-side cabinet. Protocols that route only when necessary (QL-F, cluster-based) are more robust to that choice than protocols that always transmit directly. This is the kind of robustness property that single-configuration benchmarks hide, and it matters when picking a protocol for deployment.

6.4 Qualitative Trade-Off

Table 5 sums up the trade-offs across four practical dimensions: energy efficiency, control-plane overhead, memory footprint, and adaptability to changing conditions. None of the six implementations wins on all four. QL-F has the best lifetime and very low per-packet energy, but it needs a Q-table and a warm-up period, and its behaviour during the first few hundred rounds is less predictable. SSPT gives a good all-round balance and is the only implementation that corresponds to a standardised protocol family. LEACH and EWC are simple to build and cheap in memory, but they spend a lot of energy per packet. FLAT looks surprisingly good in our setup, but its performance is fragile: change the sink position and this ranking will not hold.

Table 5
Qualitative trade-off across the six evaluated protocols.

Protocol	Energy eff.	Overhead	Memory	Adaptability
FLAT	High*	Very high	Low	Low
PEGASIS	Low	Low	Low	Low
LEACH	Medium	Medium	Low	Medium
EWC	Medium-High	Medium-High	Low	Medium
SSPT	High	Medium	Low-Medium	Medium
QL-F	**Very high**	Medium	Medium-High	**Very high**

** FLAT is energy-efficient only when the sink is near the geometric centre and the field side L is small enough that the average transmit distance stays below $d_0 \approx 87$ m.*

Limitations and Directions for Future Work

A study like this is only as useful as it is honest about its limits. Below we list ours, and for each one we point to a concrete next step.

7.1 Idealised MAC and Link Layer

The simulator does not model the MAC layer at all. Every transmission just succeeds, unless the sender has run out of energy. In reality, CSMA/CA and TSCH add retransmissions, collisions, and idle-listening, all of which cost extra energy. This extra cost may not affect all six protocols equally: relative rankings may change under realistic MAC and channel conditions, and the absolute numbers we report are optimistic. The obvious next step is to port the simulator to NS-3 or Cooja, run the same six protocols over a real IEEE 802.15.4 TSCH MAC, and see how much the numbers change.

7.2 No Channel Fading

Eqs. (1)–(3) use deterministic path loss with no shadowing or small-scale fading. In field deployments, link-quality variability is a major factor, particularly for protocols that rely on link-quality metrics (RPL with MRHOF, multipath protocols). Adding log-normal shadowing and Rayleigh/Rician fading is straightforward in the current simulator and would be a natural follow-up.

7.3 Small Network Scale

Evaluation is limited to $N \leq 200$ and $L = 100$ m. Smart-city and industrial-IoT deployments reach tens of thousands of nodes over square kilometres; at that scale, the control-plane overhead of RPL and the cluster-reformation cost of hierarchical protocols become dominant in ways the present experiments cannot reveal.

7.4 Simplified Learning-Based Protocol

QL-F is a deliberately simple Q-learning router with a tabular Q-table and an ε-greedy policy. Published learning-based routers extend this baseline with richer reward functions, multi-objective scoring, or deep neural-network function approximation [22, 23]. The QL-F numbers we report should be read as a lower bound: a properly tuned RL or DRL router would almost certainly do better.

7.5 Classical Baselines

We compare against well-known baselines (LEACH, PEGASIS) from the 2002–2012 window. We chose them because they are fully specified, deterministic to implement, and still widely cited as reference points. A parallel study against 2022–2025 learning-based proposals would be valuable, and we would expect such proposals to beat our QL-F baseline by a meaningful margin.

7.6 Field Side Length and Density

Section 6.3 covers sink position explicitly, but the orthogonal axis — field side length L — is still untested. At L = 100 m the average source-to-sink distance stays inside the cheap d^2 regime even for FLAT. At L = 200 m or more, the d^4 regime takes over for everyone, and the ranking between QL-F and the cluster-based protocols is the one we would most want to see. The simulator accepts L as a parameter, so this is a short script away.

7.7 Single Traffic Model

Only periodic reporting is evaluated. Event-driven and query-based traffic patterns favour different protocol families (TEEN/APTEEN for events; Directed Diffusion for queries) and would likely re-order the results. Similarly, we assume a static topology with fixed node positions throughout the simulation; extending the evaluation to scenarios with mobile nodes or mobile sinks — an increasingly common setting in IoT deployments [24] — would change both the energy footprint and the relative ranking of protocols.

We see this list of limits as a roadmap, not a set of excuses. Each item is a paper that someone (including us) could write next, and the open-source simulator makes that much easier to start.

Conclusions

Our goal in this paper was narrow: to give a transparent, reproducible side-by-side comparison of the main families of energy-efficient routing for IoT. The vehicle is a small Python/NumPy simulator that runs six reference implementations under the first-order radio model — FLAT, LEACH, EWC (an energy-weighted clustering scheme), PEGASIS, SSPT (a static shortest-path tree), and QL-F (a minimal baseline Q-learning implementation). In the baseline configuration with a central sink, QL-F gave the longest first-node-death time at 1411 rounds, SSPT gave the lowest per-packet energy among the non-trivial protocols at 0.97 mJ, and FLAT came second on lifetime. The sink-placement study in Section 6.3 is the more informative result: across the five sink positions tested, FLAT loses 65% of its lifetime when the

sink moves to a field corner, while QL-F's first-node-death time stays within 5% of its central-sink value. Any evaluation that fixes the sink at the centre misses this kind of robustness gap.

We want to be clear about what this work is not. EWC, SSPT, and QL-F are stand-ins for their paradigms, not faithful implementations of HEED, RPL, or any published learning-based router. The setup itself is simplified — no MAC contention, no fading, one traffic pattern, a fixed sink — and Section 7 turns each of those simplifications into an explicit next step. The simulator is open source. Anyone who wants to build on this work, swap in real protocols, or take the whole setup to NS-3 or Cooja can start from the released code.

Abbreviations

IoT	**Internet of Things**
WSN	Wireless Sensor Network
FND	First-Node-Death (time)
HND	Half-Nodes-Death (time)
LND	Last-Node-Death (time)
LEACH	Low-Energy Adaptive Clustering Hierarchy
HEED	Hybrid Energy-Efficient Distributed clustering
TEEN	Threshold sensitive Energy Efficient sensor Network protocol
SEP	Stable Election Protocol
DEEC	Distributed Energy-Efficient Clustering
GEAR	Geographic and Energy-Aware Routing
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
RPL	Routing Protocol for Low-Power and Lossy Networks
DODAG	Destination-Oriented Directed Acyclic Graph
EWC	Energy-Weighted Clustering (this paper)
SSPT	Static Shortest-Path Tree (this paper)
QL-F	Q-Learning Forwarding (this paper)
MAC	Medium Access Control
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
TSCH	Time-Slotted Channel Hopping
RL / DRL	Reinforcement Learning / Deep Reinforcement Learning

Declarations

Acknowledgments

The authors thank Sohar University for providing access to computational resources used in the simulation experiments reported in this paper.

Funding

This research received no external funding.

Conflict of Interest

The authors declare no competing interests, financial or non-financial, that could have appeared to influence the work reported in this paper.

Ethics Approval

Not applicable. This study did not involve human participants, human data, or animal subjects. All experiments are software simulations conducted on synthetic data.

Clinical Trial Number

Clinical trial number: not applicable. This study did not involve human participants, human data, or animal subjects, and no clinical trial was conducted. All experiments are software simulations performed on synthetic data.

Consent to Participate

Not applicable.

Consent for Publication

Not applicable. This manuscript does not contain any individual person's data in any form.

Availability of Data and Materials

The Python/NumPy simulator source code and the raw Monte Carlo output data supporting the findings of this study are provided as supplementary materials accompanying this manuscript. Upon acceptance, all materials will also be deposited in Zenodo with a citable DOI to ensure long-term accessibility and reproducibility. No third-party datasets were used in this study.

Code Availability

The complete source code of the Python/NumPy simulator (sink_sensitivity.py) used to produce all results reported in Sections 5 and 6 is provided as supplementary material. The code is released under

an open-source licence (MIT) to support full reproducibility. A permanent archived version with a citable DOI will be deposited at Zenodo upon acceptance.

Authors' Contributions

M.H. (Mohammad Haribat) conceived the research idea, developed the methodology, designed and implemented the Python/NumPy simulator, conducted all simulation experiments, performed formal analysis and data curation, prepared all visualisations, and drafted the original manuscript. A.K. (Ahmad Kayed) contributed to the conceptualisation, validated the experimental results and analytical interpretations, supervised the research, and reviewed and edited the manuscript. Both authors have read and approved the final version of the manuscript.

Use of Generative Artificial Intelligence

During the preparation of this manuscript, the authors used Anthropic's Claude (version Claude 4) for the purposes of minor language polishing, light editorial assistance with organising the initial outline, and generating a preliminary scaffold of the Python/NumPy simulator described in Section 5, which was subsequently rewritten and substantially modified by the authors. The authors have reviewed and edited the output and take full responsibility for the content of this publication. All intellectual contributions, including the research design, theoretical framework, mathematical derivations, simulation results, and analytical interpretations, are entirely the work of the authors. Generative AI was not used to fabricate data or invent references and is not credited with authorship.

References

1. IoT Analytics. (2024). *State of IoT 2024: number of connected IoT devices growing 13% to 18.8 billion globally*. Industry Report.
2. Dauda, A., Flauzac, O., & Nolot, F. (2024). A survey on IoT application architectures. *Sensors (Basel, Switzerland)*, *24*, 5320. https://doi.org/10.3390/s24165320
3. Kassab, W., & Darabkh, K. A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discov Internet Things*, *4*, 8. https://doi.org/10.1007/s43926-024-00084-3
4. Polastre, J., Szewczyk, R., & Culler, D. (2005). Telos: enabling ultra-low power wireless research. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, pp. 364–369.
5. Bekal, P., Kumar, P., Mane, P. R., & Prabhu, G. (2024). A comprehensive review of energy efficient routing protocols for query driven wireless sensor networks. *F1000Research*, *12*, 644. https://doi.org/10.12688/f1000research.133874.2
6. Bomnale, A., & More, A. (2025). A survey of data aggregation and routing protocols for energy-efficient wireless sensor networks. *EAI Endorsed Trans Scal Inf Syst*, *12*(2), e6924. https://doi.org/10.4108/eetsis.6924

7. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans Wirel Commun*, *1*, 660–670. https://doi.org/10.1109/TWC.2002.804190
8. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future Internet: the Internet of Things architecture, possible applications and key challenges. In: Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, pp. 257–260.
9. Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *Ieee Signal Processing Magazine*, *19*, 40–50.
10. Ogundile, O. O., & Alfa, A. S. (2017). A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks. *Sensors (Basel, Switzerland)*, *17*, 1084. https://doi.org/10.3390/s17051084
11. Shafiq, M., Ashraf, H., Ullah, A., & Tahira, S. (2020). Systematic literature review on energy efficient routing schemes in WSN — a survey. *Mob Netw Appl*, *25*(3), 882–895. https://doi.org/10.1007/s11036-020-01523-5
12. Thakur, S., Sarkar, N. I., & Yongchareon, S. (2025). AI-driven energy-efficient routing in IoT-based wireless sensor networks: a comprehensive review. *Sensors (Basel, Switzerland)*, *25*(24), 7408. https://doi.org/10.3390/s25247408
13. Del-Valle-Soto, C., Rodríguez, A., & Ascencio-Piña, C. R. (2023). A survey of energy-efficient clustering routing protocols for wireless sensor networks based on metaheuristic approaches. *Artificial Intelligence Review*, *56*, 9699–9770. https://doi.org/10.1007/s10462-023-10402-w
14. Amutha, J., Sharma, S., & Nagar, J. (2020). WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: review, approaches and open issues. *Wirel Pers Commun*, *111*, 1089–1115. https://doi.org/10.1007/s11277-019-06903-z
15. Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Ieee Transactions On Mobile Computing*, *3*, 366–379.
16. Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, CA, USA, pp. 2009–2015.
17. Smaragdakis, G., Matta, I., & Bestavros, A. (2004). SEP: a stable election protocol for clustered heterogeneous wireless sensor networks. In: Proceedings of the 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA), Boston, MA, USA.
18. Qing, L., Zhu, Q., & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer Communications*, *29*, 2230–2237.
19. Yu, Y., Govindan, R., & Estrin, D. (2001). Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. UCLA CSD Technical Report UCLA/CSD-TR-01-0023, University of California, Los Angeles, USA.
20. Lindsey, S., & Raghavendra, C. S. (2002). PEGASIS: power-efficient gathering in sensor information systems. In: Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, vol. 3, pp. 1125–1130

https://doi.org/10.1109/AERO.2002.1035242

21. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J. P., & Alexander, R. (Eds.). (2012). : RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550.
22. Mutombo, V. K., Lee, S., Lee, J., & Hong, J. (2021). EER-RL: energy-efficient routing based on reinforcement learning. Mob. Inf. Syst. 5589145 (2021). https://doi.org/10.1155/2021/5589145
23. Sharma, H., Haque, A., & Blaabjerg, F. (2021). Machine learning in wireless sensor networks for smart cities: a survey. *Electronics*, *10*, 1012. https://doi.org/10.3390/electronics10091012
24. Temene, N., Sergiou, C., Georgiou, C., & Vassiliou, V. (2022). A survey on mobility in wireless sensor networks. *Ad Hoc Networks*, *125*, 102726. https://doi.org/10.1016/j.adhoc.2021.102726
25. Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report EBSE-2007-01, Keele University and Durham University Joint Report, Keele and Durham, UK.
26. Medeiros, D. F., de Souza, C. P., de Carvalho, F. B. S., & Lopes, W. T. A. (2022). Energy-saving routing protocols for smart cities. *Energies*, *15*, 7382. https://doi.org/10.3390/en15197382
27. Yao, Y. D., Li, X., Cui, Y. P., Wang, J. J., & Wang, C. (2022). Energy-efficient routing protocol based on multi-threshold segmentation in wireless sensor networks for precision agriculture. *Ieee Sensors Journal*, *22*(7), 6216–6231. https://doi.org/10.1109/JSEN.2022.3150770
28. Darabkh, K. A., & Al-Akhras, M. (2025). Evolutionary cost analysis and computational intelligence for energy efficiency in Internet of Things-enabled smart cities: multi-sensor data fusion and resilience to link and device failures. *Smart Cities*, *8*(2), 64. https://doi.org/10.3390/smartcities8020064

Figures

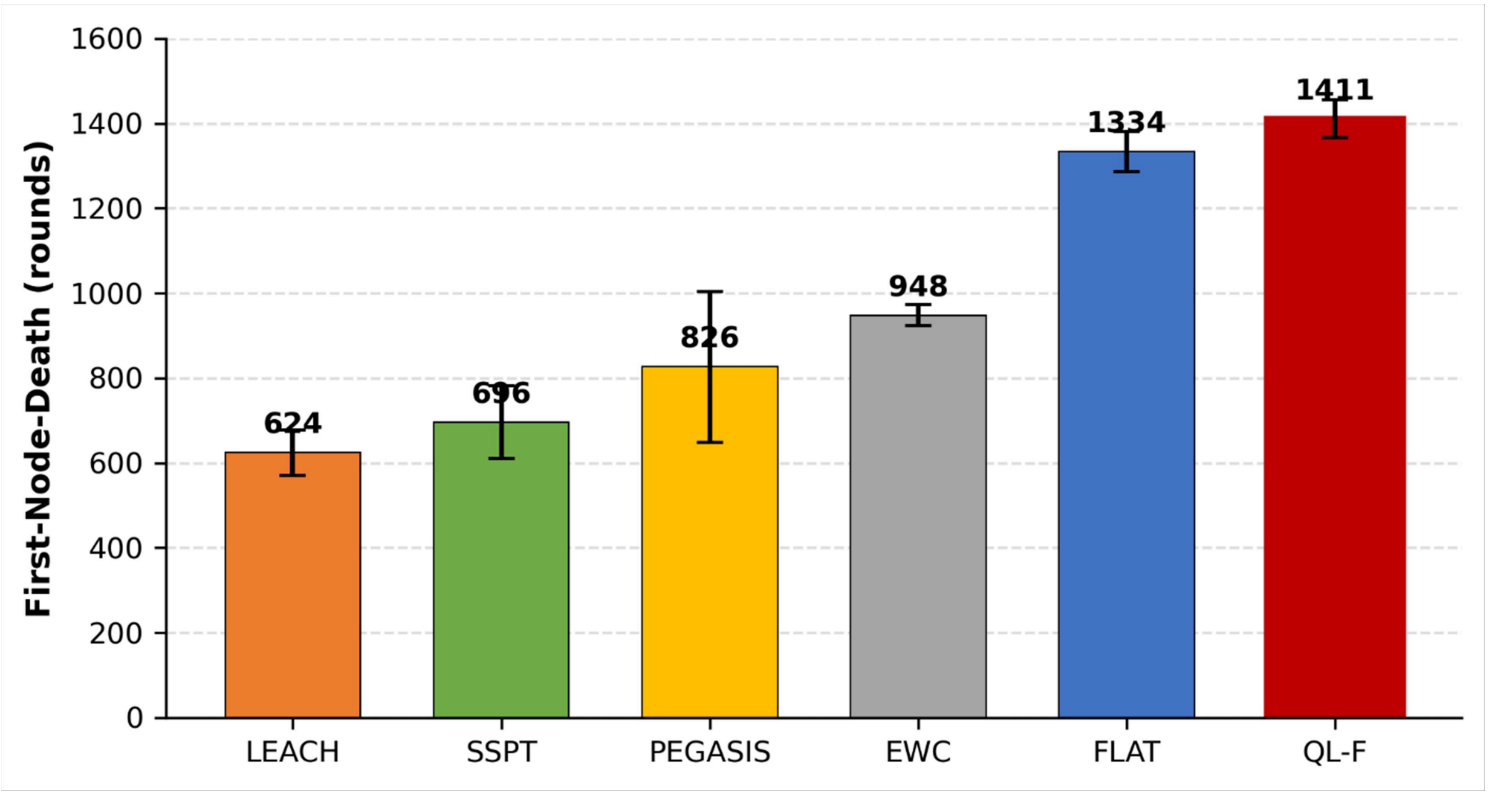


Figure 1

First-Node-Death time at N = 100 for the six protocols. Error bars show one standard deviation across 30 runs.

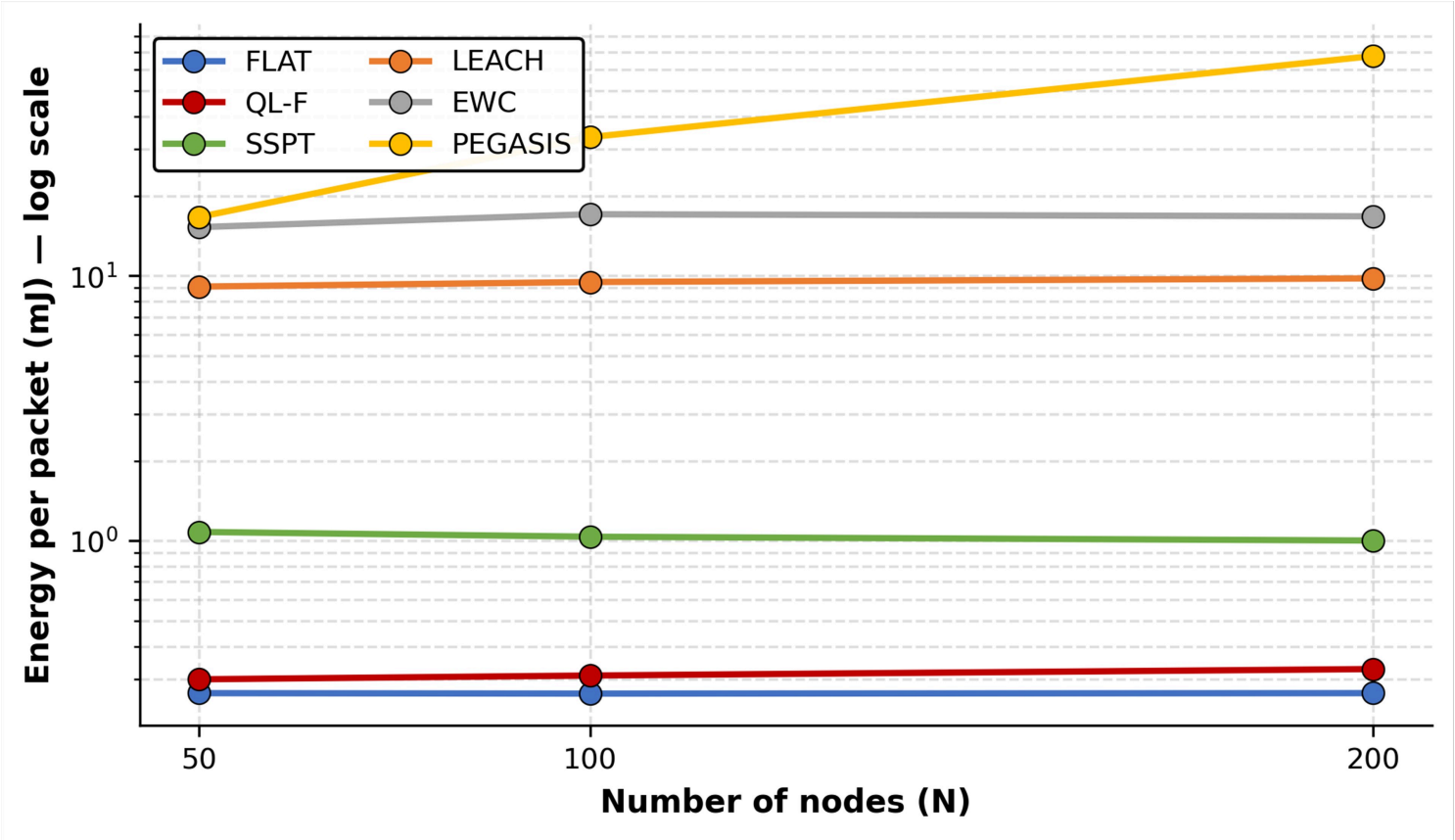


Figure 2

Per-packet energy as N grows from 50 to 200 (log scale). FLAT and QL-F stay under 0.35 mJ, while PEGASIS grows roughly linearly with N.

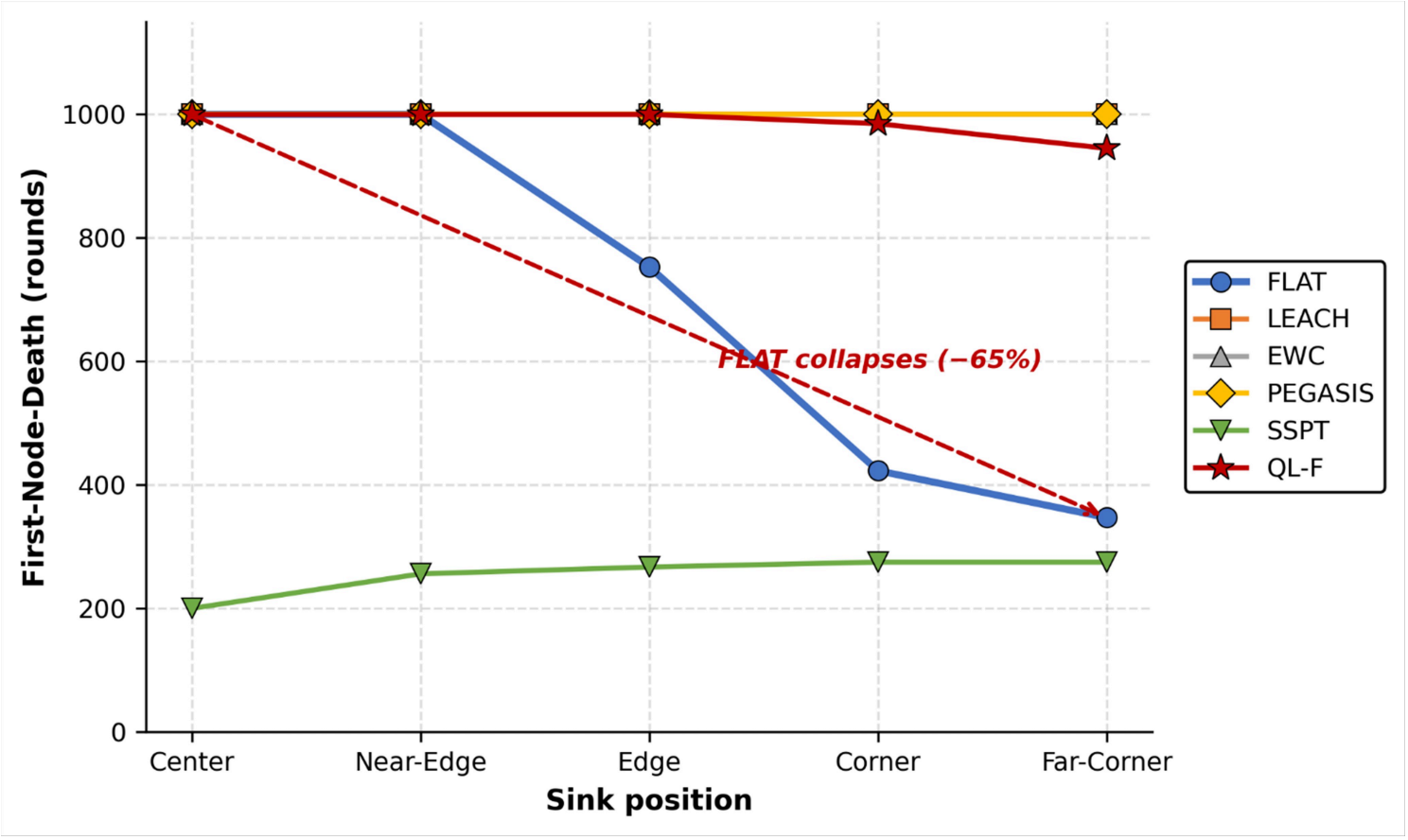


Figure 3

First-Node-Death as the sink moves from field centre to far corner. FLAT loses about 65% of its lifetime; across the five sink positions tested, QL-F's first-node-death time stays within 5% of its central-sink value.