Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

DASE-IoT Deadline-Aware and Energy-Balanced Computation Offloading across Device, Edge, and Cloud

Mr. Pawan Kumar
Assistant Professor
Department of Amity Institute of Information Technology
Amity University Raipur Chhattisgarh
pkumar@rpr.amity.edu

ABSTRACT

Internet of Things (IoT) applications increasingly require decisions about where a computational task should run: on a battery-limited device, at a nearby edge node, or in a remote cloud. Minimizing execution time alone can waste device energy, while minimizing energy alone can violate application deadlines. This paper presents DASE-IoT, a lightweight deadline-aware scheduling policy that first identifies execution alternatives predicted to satisfy the task deadline and then balances normalized device energy against response time. A closed-form three-tier analytical model captures CPU work, uplink and downlink transmission, queueing, and cloud backhaul delay. The policy is evaluated with 25,000 synthetic tasks across ten independently seeded runs. Under the stated assumptions, DASE-IoT satisfies 90.2% of task deadlines while reducing modeled device energy by 10.4% compared with a latency-first scheduler, with a 3.3% mean-latency increase. The results establish a reproducible feasibility demonstration, not a hardware-validated performance claim.

Keywords: Internet of Things, edge computing, computation offloading, deadline-aware scheduling, energy efficiency, simulation.

1. Introduction

Connected sensors, wearables, industrial controllers, and smart-building devices transform continuous measurements into alerts or computational decisions. The Internet of Things combines sensing, networking, and data processing across heterogeneous devices [1]. While many endpoint processors can handle small tasks locally, larger workloads may be redirected to nearby edge servers or cloud platforms. Edge computing is motivated partly by the need for

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

responsive processing, reduced bandwidth pressure, and application locality [2], [3]. The location of execution, however, changes both response time and the energy paid by the originating device.

A device-side inference request illustrates the tension. Local execution avoids radio transfer but may consume substantial CPU energy and finish too late. An edge server may compute quickly, yet the task must first cross a wireless uplink and wait in a queue. A remote cloud provides additional computational capacity but introduces network and backhaul delays. Therefore, an apparently faster processor need not deliver the fastest completed task. Wireless conditions and workload sizes also vary across time, so a fixed rule such as “always use edge” is rarely appropriate.

1.1 Research problem

For each independent task i, the scheduler observes an estimated workload, input/output data sizes, radio rates, candidate execution times, device-energy costs, and a latency deadline. It must select exactly one of three destinations: local device (L), edge (E), or cloud (C). The primary requirement is to satisfy the deadline whenever a modeled feasible destination exists. Within that constraint, the secondary aim is to balance device energy consumption with timely completion. The design is intentionally small enough to run as a per-task decision rule without iterative training or centralized optimization.

1.2 Contributions

- A compact analytical model quantifies local, edge, and cloud task latency and endpoint energy on common units.
- A deadline-filtered weighted scoring policy is specified with an explicit fallback for tasks that cannot meet their deadline.
- A reproducible seeded simulation compares the policy against local-only, edge-only, cloud-only, latency-first, and energy-first alternatives.
- A queue-congestion sensitivity case and explicit limitations clarify what the initial results support.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

1.3 Research question and scope

The central research question is: Can a simple deadline-feasibility filter preserve the highest attainable modeled deadline satisfaction while reducing endpoint energy relative to choosing the fastest destination? The answer is examined under a tractable assumption: true simulated per-task costs are available to the scheduler at decision time. Consequently, this study measures scheduling behavior under model-perfect information, rather than demonstrating prediction robustness or deployed quality of service.

2. Related Work and Research Gap

2.1 IoT processing continuum

Atzori et al. [1] provided an early conceptual synthesis of connected objects, enabling technologies, and system-level challenges. Shi et al. [2] developed the edge-computing vision for placing processing closer to data producers, while Satyanarayanan [3] argued that edge placement can improve responsiveness and resilience for mobile and IoT workloads. Together, these works motivate a device-edge-cloud continuum rather than treating the cloud as the sole destination.

2.2 Computation offloading and resource trade-offs

Mao et al. [4] reviewed mobile edge computing from the communications perspective and stressed the coupling between radio and computational resource allocation. Mach and Becvar [5] characterized the principal offloading decisions and architectural considerations. Later reviews explicitly organized optimization strategies by energy, quality of service, and other constraints [6], [7]. These studies show that offloading is not a binary question: transfer cost, compute capacity, and queueing together determine whether remote execution is useful.

2.3 Current research direction

Recent IoT-oriented work studies multi-access edge task offloading, including cost-sensitive and latency-sensitive policies [8], [9]. Sophisticated methods can account for interference, shared resources, and learned system dynamics. Such methods may outperform a simple heuristic, but their improvements can be difficult to interpret without a transparent lower-complexity reference. This paper does not claim to supersede learned or globally optimized

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

schedulers. Instead, it investigates a fully specified and readily auditable baseline that prioritizes deadline feasibility before selecting among possible processing locations.

2.4 Positioning of this study

Prior direction	**What is emphasized**	**DASE-IoT emphasis**
IoT and edge architectures [1]–[3]	Why computation should move nearer to devices	Per-task choice among three tiers
MEC surveys [4], [5]	Joint communication/computation constraints	Explicit time and endpoint-energy equations
Recent offloading reviews [6]–[9]	Algorithm families and quality-of-service trade-offs	Small interpretable rule and repeatable benchmark

Table 1. Positioning of the proposed proof-of-concept relative to cited literature.

The research gap addressed here is methodological rather than a claim of universal algorithmic novelty: a concise three-tier baseline, the same synthetic tasks for every comparator, published simulation parameters, and precise interpretation of deadline guarantees. Security, service placement, inter-task precedence, and network-wide optimization are outside the paper's evaluation scope; in practical deployments they are important dimensions of a complete offloading system.

3. Architecture and Mathematical Model

The considered IoT node profiles each task and queries candidate execution estimates. A small decision controller selects local, edge, or cloud execution. The controller may physically reside on the device or its gateway; the present cost model excludes the controller's own runtime overhead. Each task is indivisible and assigned to exactly one destination.

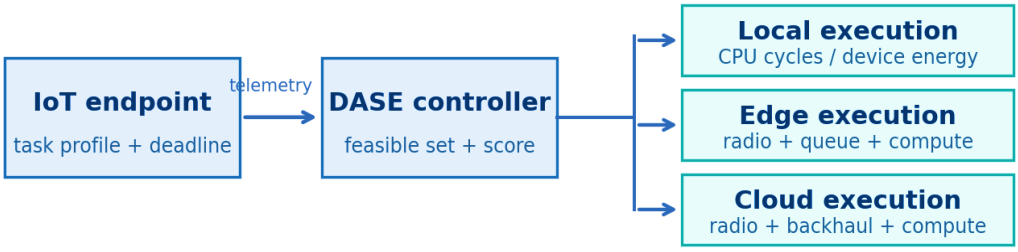


Fig. 1. Three-tier task-placement architecture and per-task decision flow.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

3.1 Task representation

Task i is represented by (C_i, B_i, O_i, D_i), denoting required CPU cycles, input bits, output bits, and deadline. Device, edge, and cloud computing rates are fL, fE, and fC; uplink/downlink rates are Ru and Rd. The estimated queueing delays are qE and qC, while b and h denote cloud backhaul and additional transport delay. All latencies are measured in seconds.

$$T(L) = C/fL$$

$$T(E) = B/Ru + O/Rd + C/fE + qE$$

$$T(C) = B/Ru + O/Rd + C/fC + qC + b + h$$

3.2 Endpoint energy expenditure

Local dynamic CPU energy follows a standard frequency-squared analytical approximation, with device-dependent coefficient κ. For remote execution, the device spends energy sending and receiving data and remains in an idle-power state while waiting. These expressions describe originating-device energy only: they do not measure edge/cloud facility energy or embodied impacts.

$$E(L) = \kappa \cdot C \cdot fL^2$$

$$E(k) = Ptx \cdot B/Ru + Prx \cdot O/Rd + Pidle \cdot [T(k) - B/Ru - O/Rd], \quad k \in \{E, C\}$$

3.3 Feasibility boundary

An option k is feasible when $T(k) \leq D$. This deterministic condition depends on estimates that may be wrong in a deployment. The model assumes nonoverlapping radio and compute intervals, consistent measurement units, and no contention effects beyond the injected queue delay. A task with no feasible option is counted as a deadline miss regardless of the selected placement.

4. Proposed DASE-IoT Scheduling Policy

The deadline-aware scheduling policy first constructs the feasible set $F = \{k \in \{L, E, C\}: T(k) \leq D\}$. Within F, it minimizes a positive weighted sum of normalized energy and normalized latency. For this proof of concept, $\alpha = 0.65$ weights the endpoint-energy term. The energy term is normalized to the corresponding local execution energy, and latency is normalized to the

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

task deadline. When F is empty, the method selects the smallest predicted execution time and records an unavoidable modeled deadline miss.

$$J(k) = \alpha \cdot E(k)/\max[E(L), 0.03\ J] + (1 - \alpha) \cdot T(k)/D, \quad \alpha = 0.65$$

$$k^* = \arg\min[k \in F]\ J(k) \quad \text{if } F \neq \emptyset; \quad \text{otherwise} \quad k^* = \arg\min[k \in \{L,E,C\}]\ T(k).$$

4.1 Decision procedure

INPUT Task features and estimated T(k), E(k) for k = L, E, C.

1. Compute the feasible set F from $T(k) \leq D$.
2. If F is nonempty, calculate J(k) for each k in F.
3. Choose the feasible k with minimum J(k); break ties by lower T(k).
4. Otherwise choose the fastest k and flag a deadline violation.
5. Return the placement decision and predicted cost.

With exactly three candidates, evaluation takes O(3) comparisons and constant storage per independent task. Positive scoring weights imply that the selected feasible choice is not strictly dominated on both modeled energy and latency within the candidate set. This property does not guarantee a globally optimal network assignment when multiple devices compete for shared resources.

5. Experimental Design

The numerical study draws 2,500 independent tasks per run and repeats the same experiment under ten seeds (2026–2035), for 25,000 simulated tasks in total. All algorithms evaluate identical tasks within a seed. The implementation uses NumPy; there is no hardware testbed and no publicly collected device trace.

Parameter	**Synthetic distribution / fixed setting**
Compute demand C; input B	Uniform 0.2–1.4 × 10^9 cycles; 0.15–2.5 MB
Device / edge / cloud CPU	Uniform 0.45–1.0 / 2.2–5.0 / 6–11 GHz
Uplink / downlink rates	Uniform 4–18 / 9–28 Mb/s
Edge / cloud queue delay	Uniform 15–180 / 20–100 ms
Cloud backhaul / extra delay	Uniform 60–140 / 0–130 ms

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

Power and coefficient	Tx 0.70 W; Rx 0.35 W; idle 0.08 W; $\kappa = 0.8\times10^{-27}$
Response output; deadlines	1–6% of input; D = U(0.68,1.85)·min(T) + U(0.08,0.26) s

Table 2. Synthetic assumptions; uniform draws are independent unless otherwise stated.

6. Results and Discussion

Table 3 reports means across ten seeded scenarios. "Deadlines met" is the fraction of tasks with the selected response time at or below the sampled deadline. The energy metric is average originating-device joules per task, not network-wide electricity consumption. Latency-first selects the minimum modeled response time; energy-first chooses the minimum modeled endpoint energy without enforcing a deadline.

Policy	Mean latency (s)	Device energy (J)	Deadlines met (%)
DASE-IoT (proposed)	0.949	0.341	90.23
Latency-first	0.918	0.381	90.23
Energy-first	1.048	0.307	80.38
Local-only	1.161	0.354	69.34
Edge-only	1.486	0.826	51.66
Cloud-only	1.475	0.825	52.79

Table 3. Mean performance over 10 independent 2,500-task simulation runs (25,000 tasks total).

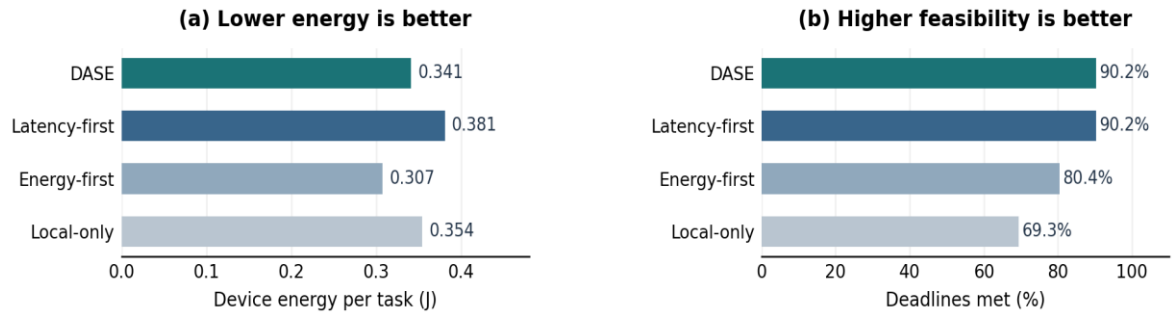


Fig. 2. Endpoint energy and on-time completion under the nominal synthetic setting.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-11 | November - 2025 | Page 65-73

6.1 Key observations

DASE-IoT reduces modeled endpoint energy by 10.4% versus latency-first while increasing mean response time from 0.918 s to 0.949 s (3.3%). Both attain 90.23% deadline satisfaction because both use exact candidate-cost information: whenever any option is feasible, the fastest option is feasible and DASE-IoT also selects a feasible option. The proposed policy does not improve the deadline-feasibility ceiling; it instead conserves endpoint energy within that ceiling. Relative to energy-first, DASE-IoT meets approximately 9.85 additional percentage points of deadlines at higher energy cost.

6.2 Edge congestion sensitivity

Multiplying sampled edge queue delays by 2.5 reduces edge selection by DASE-IoT from 12.05% to 3.62%, while cloud selection increases from 21.00% to 28.60%. Deadline satisfaction changes only from 90.23% to 90.11% under the same seeded experimental protocol. These values reflect modeled adaptation, not a measured resilience guarantee. A repeatable sensitivity test is more meaningful than a claim that the scheduler is always superior: on the synthetic workload, energy-first consumes the least device energy, and latency-first always achieves the smallest mean task latency.

7. Conclusion, Limitations and Future Work

This paper introduced DASE-IoT, a transparent scheduling policy for routing independent IoT workloads among device, edge, and cloud execution. A feasibility-first decision rule retains the attainable modeled on-time completion rate while balancing normalized latency and endpoint energy. In 25,000 synthetic tasks, the method achieved 90.23% deadlines met and 10.4% lower modeled device energy than a latency-first policy, at the cost of 3.3% higher mean latency. These are results of the specified synthetic model, not field measurements or evidence of superiority to all existing offloading approaches.

The study assumes perfect cost estimates, independent tasks, a simple idle-power model, and externally supplied queue delays. It does not jointly allocate wireless bandwidth or compute resources, model packet loss, account for the energy consumed by edge/cloud servers, or test data sensitivity and security-policy constraints. Future work should evaluate prediction error, multi-device contention, realistic network traces, dynamic voltage/frequency scaling, fairness, and physical energy measurements on IoT gateways. Broader validation and comparison with

state-of-the-art optimization or learning methods are necessary before submission as a mature experimental journal contribution.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010. doi: 10.1016/j.comnet.2010.05.010.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016. doi: 10.1109/JIOT.2016.2579198.

[3] M. Satyanarayanan, "The Emergence of Edge Computing," Computer, vol. 50, no. 1, pp. 30–39, 2017. doi: 10.1109/MC.2017.9.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017. doi: 10.1109/COMST.2017.2745201.

[5] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, 2017. doi: 10.1109/COMST.2017.2682318.

[6] J. Yang, A. A. Shah, and D. Pezaros, "A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks," Electronics, vol. 12, no. 17, art. 3548, 2023. doi: 10.3390/electronics12173548.

[7] Kanupriya, I. Chana, and R. K. Goyal, "Computation offloading techniques in edge computing: A systematic review based on energy, QoS and authentication," Concurrency and Computation: Practice and Experience, vol. 36, no. 13, e8050, 2024. doi: 10.1002/cpe.8050.

[8] D. Wang, K. B. A. Bakar, B. Isyaku, T. A. E. Eisa, and A. Abdelmaboud, "A comprehensive review on internet of things task offloading in multi-access edge computing," Heliyon, vol. 10, no. 9, e29916, 2024. doi: 10.1016/j.heliyon.2024.e29916.

[9] C. Eang, S. Ros, S. Kang, I. Song, P. Tam, S. Math, and S. Kim, "Offloading Decision and Resource Allocation in Mobile Edge Computing for Cost and Latency Efficiencies in Real-Time IoT," Electronics, vol. 13, no. 7, art. 1218, 2024. doi: 10.3390/electronics13071218.