AOI: 10.100.234513.0269
13th international conference on Microelectronics, Circuits and Systems, Micro2026

Energy-Aware Priority-Based Task Scheduling in IoT–Network

Shaad Ali, Zaineb Naaz, Vidushi Sharma, Vivek Chaudhary

Department of Computer Science and Engineering
Gautam Buddha University, Greater Noida, Uttar Pradesh, India.
Corresponding Author: vivek.chaudhary@gbu.ac.in

ABSTRACT

The rapid emergence of Internet of Things (IoT) applications has made it even more imperative to schedule tasks efficiently. The traditional algorithms, e.g., First-Come-First-Serve (FCFS) and Earliest Deadline First (EDF) algorithms are concerned mainly with the time ordering of tasks, and do not take into account the energy constraints and varying load conditions in hierarchical systems. In this paper, we present an energy-efficient and priority-based task scheduling framework for hierarchical IoT networks that include edge, fog and cloud layers. The aim of the proposed scheme is to balance queue load across hierarchical IoT layers while considering edge node energy consumption and task priority to ensure efficient and reliable task scheduling. To model task arrivals and service processes under varying load conditions, M/M/1 and M/M/c queueing models are employed for performance analysis and adaptive scheduling decisions. Simulation results demonstrate that the proposed approach outperforms FCFS and EDF in terms of reduced response time, energy consumption, and enhanced overall system efficiency. These findings highlight the effectiveness of integrating hierarchical task management, queueing analysis, and adaptive priority scheduling in modern IoT systems.

Keywords: Internet of Things, Energy-aware scheduling, Priority-based task scheduling, Load balancing, Service Level Agreement.

I. INTRODUCTION

The rapid emergence of Internet of Things (IoT) applications has resulted in a massive amount of data being generated and a large number of computational tasks that must be processed at a faster pace. The cloud-based architecture, which is commonly used, involves large communication delays and excessive energy consumption due to the transmission of data over a large distance. To overcome these issues, hierarchical IoT-Network architectures have been developed. In such multi-layer scenarios, effective task scheduling is a critical task. The conventional task scheduling algorithms such as FCFS and Earliest Deadline First EDF primarily focus on the order in which tasks are executed. However, such conventional approaches fail to consider the energy constraints of edge devices, the dynamic nature of the queue, and the imbalanced task distribution over the layers. Since IoT devices have relatively shorter battery lives, energy-efficient task scheduling is necessary to ensure the reliability of the system.

There are already a number of studies on scheduling that consider latency and deadlines in fog and cloud scenario. However, limited attention has been given to integrated frameworks that simultaneously address priority handling, queue dynamics, and residual energy constraints in IoT network. In real-world IoT deployments, if these aspects are not taken into account, it may lead to excessive energy usage, violation of SLAs, and reduction in system throughput.

To address these challenges, this paper presents an energy-aware, priority-based scheduling framework for hierarchical IoT–network. The proposed model incorporates queue-aware load balancing, adaptive priority control, and energy-constrained task admission at the edge layer. A mathematical queueing model based on M/M/1 (edge/fog) and M/M/c (cloud) systems is used to analyze system behavior. Monte Carlo simulations are conducted to evaluate system performance under varying workload conditions.

The main contributions of this work are:

- Developing a hierarchical scheduling framework that considers energy constraints of edge and fog nodes.
- Adaptive load balancing using queue length, utilization, and residual energy.
- Monte Carlo simulations that include load sensitivity analysis are used to test statistical performance.

The rest of this paper is structured as follows Section II looks at work that is related. The system model is shown in Section III. The proposed scheduling framework is explained in Section IV. Section V talks about how the experiment was set up and how well it worked. Section VI is about Conclusion.

II. RELATED WORK

Extensive research has been conducted in recent years on task scheduling and workload management in hierarchical IoT network to address latency, congestion, and energy constraints [1], [2], [3], [4]. Bali et al. [5] designed a Priority-Aware Task Scheduling (PaTS) scheme using multi-level queues to distinguish between emergency and non-emergency tasks, and a multi objective NSGA-II approach to enhance performance. Their scheme reduces latency and energy consumption more efficiently, but it does not consider dynamic

ISBN: 978-81-689089-0-1

modelling of energy depletion and recovery processes at edge nodes.

Sonmez et al. [6] designed a fuzzy logic-based orchestration scheme for multi-tier edge computing environments. Their approach analyses the optimal execution tiers based on server utilization, network quality, and task types. Their scheme is helpful in dynamic environments, but it is more focused on orchestration strategy and does not provide rigorous queue theoretic analysis or energy-aware scheduling constraints.

Naaz and Sharma [7] analysed the performance of edge nodes under the growing demand of IoT tasks, particularly in terms of processing delay and energy consumption. Their study highlights congestion problems in edge tiers and the need for adaptive scheduling techniques. Further, queue-theoretic approaches have been extensively studied by the researchers. For instance, Zhang et al. [8] proposed a delay and load fairness optimization model for multi-access edge computing based on queueing theory. Their model considers the balance between service delay and fairness among nodes, but it does not consider energy control approaches that are priority-sensitive. In IoT-Fog-Cloud systems, there have also been studies on priority-based queueing policies [9],[10]. These policies have been demonstrated to be effective in improving the response time of high-priority tasks, but they typically overlook long-term energy feasibility.

Game-theoretic load management approaches such as LMGT [11] address fairness and energy efficiency by carefully distributing the workload among edge nodes. These approaches are effective in handling fairness, but they typically require complex optimization procedures and do not directly examine statistical robustness when tasks are randomly received.

Recent surveys on task offloading in edge computing [12] indicate that most existing approaches consider latency, energy efficiency, and reliability as independent objectives that are only loosely coupled. The authors in [13] Delay-aware scheduling architectures they aim to make systems more reactive in real-time, but they do not typically integrate priority management, energy simulation, and statistical confidence validation into a single framework.

Even though there has been a lot of progress in priority based scheduling, queue-theoretic modelling, fuzzy orchestration, and game-theoretic load balancing, most current methods still treat latency, energy efficiency, and workload fairness as separate problems to solve. Many models either try to cut down on delays or divide up resources, but they don't have ways for edge nodes to lose and regain energy. Furthermore, while queueing theory has been applied to analyze delay and fairness, there has been inadequate emphasis on the amalgamation of priority-aware admission control with energy-sensitive decision policies in the context of stochastic workload variations. Statistical robustness analysis is not a common component of scheduling frameworks, particularly regarding the execution of multiple Monte Carlo simulations and the verification of confidence intervals. As a result, there is still not enough research on a cohesive framework that addresses priority differentiation, energy-efficient node management, adaptive load balancing, and statistical performance validation all at the same time.

III. SYSTEM MODEL

The proposed work considers a four-layer network architecture that includes an IoT layer, edge layer, fog layer, and cloud layer. The IoT layer consists of IoT devices, which are resource-constrained, and these devices generate tasks that require a high amount of processing power. If there are too many of these tasks, they can be handled locally, sent to nearby edge nodes, or sent to the cloud layer. The edge layer consists of the edge nodes, which have higher computing power than IoT devices. They are mainly located at the edge of the IoT network. The fog layer consists of fog nodes, which are positioned farther from IoT devices than edge nodes and possess higher computational power than edge nodes. Finally, the cloud layer consists of high-performance data centers with abundant resources to process large-scale workloads.

Let λ be the average speed at which IoT devices send tasks. The edge nodes are modeled as single-server systems with a service rate of μ_e, the fog layer is also modeled as intermediate processing nodes for load balancing, while the cloud layer is modeled as a multi-server system with a service rate of μ_c, and c servers that work at the same time. Each edge node maintains a queue of incoming tasks and processes them sequentially using a priority-based scheduling mechanism. Tasks are classified into two categories: high-priority and normal-priority, based on application requirements and deadline sensitivity.

The edge layer is modelled as an M/M/1 queue, where task arrivals follow a Poisson process and service times are exponentially distributed. When an edge node becomes overloaded or energy-constrained, tasks are offloaded to the fog layer. Similarly, fog nodes are also modelled as M/M/1 queues, where they perform intermediate processing and load balancing before forwarding tasks to the cloud layer, if necessary. The average system utilization at the edge node is expressed as following:

$$\rho_e = \frac{\lambda}{\mu_e}, \quad \rho_e < 1 \tag{1}$$

The average waiting time at the edge node is computed

ISBN: 978-81-689089-0-1

as follows:

$$W_e = \frac{1}{\mu_e - \lambda} \quad (2)$$

Where c represents the number of servers, and μ_c is the service rate of each server.

This hierarchical modelling allows analytical characterization of response time across layers. The edge node consumes energy for data transmission and reception, as well as task computation. The transmission energy required to send a task of size L bits over distance d follows the first-order radio energy model [14]:

$$E_{tx} = LE_{elec} + L \in_{amp} d^2 \quad (4)$$

Where E_elec represents electronic energy consumption and ∈_amp is the amplifier energy coefficient.

The computational energy required to execute a task with C CPU cycles at frequency fis given by the standard dynamic CMOS power model [15], as shown in Equation (5).

$$E_{comp} = \kappa C f^2 \quad (5)$$

Where κ is the effective switched capacitance coefficient.

The total energy consumption per task at the edge node is computed using Equation (6).

$$E_{total} = E_{tx} + E_{comp} \quad (6)$$

Furthermore, to ensure sustainability, an energy threshold $E_{\min}$ is introduced. If the residual energy falls below$E_{\min}$, the node temporarily stops receiving new tasks and enters a recovery phase.

IV. PROPOSED ENERGY-AWARE PRIORITY SCHEDULING FRAMEWORK

The proposed framework uses a hierarchical task scheduling system that works from Edge, Fog, to Cloud. The algorithm performs load balancing based on the queue, energy-based admission control, and priority-based task mapping.

A. Task Characteristics

In the proposed framework, each task T_I is defined by the following parameters:

- Arrival time a_I
- CPU cycles C_I
- Data size L_I
- Deadline T_I
- Priority level $P_I \in \{1,2,3,4\}$

Here, priority level 4 represents critical tasks that require immediate processing.

B. Energy-Aware Edge Selection

At the edge layer, each edge node continuously monitors the following parameters:

- Queue length Q_i
- Utilization ρ_i
- Residual energy ratio $E_i = \frac{E_{res}}{E_{initial}}$

An edge node turns inactive when:

$$E_i < 0.25 \quad \text{or} \quad \rho_i > 0.85 \quad (7)$$

In the proposed framework, inactive nodes go to sleep for a fixed recovery period. After waking up, 10% of the initial energy level is replenished. For active nodes, a combined load balancing score is calculated as given in Equation (9).

$$Score_i = 0.35Q_i + 0.30\rho_i + 0.35(1 - E_i) - 0.25\frac{P_i}{4} \quad (8)$$

The combined score $Score_i$ is a normalized, weighted, multi-objective decision criterion. Each parameter is normalized to the range [0,1]. The queue length Q_i indicates the current workload at the node, while the utilization ρ_i reflects the extent of resource usage. The term $(1 - E_i)$ penalizes nodes with lower residual energy, which encourages energy-aware task allocation. The priority term P_i is negatively weighted to give higher priority to tasks with higher priority values, making it more likely that they will be selected. The weight coefficients (0.35, 0.30, 0.35, 0.25) in the linear weighted aggregation method are determined using simulation-based sensitivity analysis to identify the optimal combination for balancing the trade-off between delay minimization and energy consumption. The task is assigned to the edge node with the minimum score, if:

- The node is active
- Capacity constraint is satisfied
- Residual energy is sufficient for estimated task energy

C. Fog Layer Load Balancing

If no edge node can accept the task, the fog layer is evaluated using a weighted load metric as given in Equation (9).

$$Score_{fog} = (0.6 * \frac{Q_i}{Capacity_i}) + (0.4 * \rho_i) \quad (9)$$

where $Score_{fog}$represents the combined load metric used for fog node selection, $Q_i$denotes the queue length of thei^{th}fog node, Capaciti represents its maximum processing capacity, the term $\frac{Q_i}{Capacity_i}$indicates the normalized queue load, and ρ_i represents the utilization of the i^{th} fog node defined as the ratio of arrival rate to service rate. The coefficients 0.6 and 0.4 are weight factors that balance the effect of queue load and utilization in the selection process. Fog nodes exceeding utilization threshold or with insufficient energy are skipped. The minimum-score fog node is selected.

ISBN: 978-81-689089-0-1

D. Cloud Fallback Mechanism

The task is moved to the cloud layer if both the edge and fog layers are too busy or don't have enough power to do it.

The cloud is set up as an M/M/c queue with many servers that can work at the same time and no limit on energy.

E. Priority-Based Queue Processing

Within each node, task execution follows:

- EDF policy for EDF baseline,
- Arrival-based ordering for FCFS,
- Dynamic priority ordering for the proposed scheme

.

A good priority adjustment is made based on how much time is left on the deadline, which gives urgent tasks a temporary boost in priority.

F. Energy Recovery Mechanism

When a node enters sleep mode due to low energy or prolonged inactivity, it remains inactive for a predefined recovery period. Upon reactivation, the residual energy of the node is updated as follows:

$$E_{res} = \min((E_{res} + (0.10 * E_{initial})), E_{initial}) \quad (10)$$

Where E_{res} represents the residual energy of the node after recovery and $E_{initial}$ is the initial energy capacity.

This recovery mechanism ensures that the node regains 10% of its initial energy upon reactivation without exceeding its maximum energy limit. It prevents permanent energy depletion and supports the long-term sustainability of edge resources.

V. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

The proposed framework is simulated in Python where edge layer and fog layers considers M/M/1 model and cloud layer is assumed to have M/M/c model. Table 1 presents the simulation parameters used. The performance of the proposed approach is compared with FCFS and EDF scheduling approaches. The energy model takes into account both transmission and computation energy. The cloud is thought to have unlimited energy, while fog nodes can hold 2000 Joule and edge nodes can only hold Only 100 Joule of energy.

A. Performance Metrics

To validate the performance of the proposed approach four performance metrics are considered.

- **Average Waiting Time:** This is the time that a task spends in the queue before it is completed.
- **Deadline Violation Ratio:** This is the percentage of tasks that have violated their deadlines.
- **Important SLA Satisfaction:** This is the percentage of important tasks that have been completed on time (priority level 4).
- **Total Energy Consumption:** This is the total energy consumption of the edge, fog, and cloud environments.

TABLE 1
SIMULATION PARAMETER

Parameter	Value
Edge Nodes	4
Fog Nodes	4
Cloud Servers	5
Tasks per Run	2000
Monte Carlo Runs	15
Arrival Process	Exponential (Poisson)
Arrival Scale	0.18 (baseline)
Edge Service Scale	4.0
Fog Service Scale	2.0
Cloud Service Scale	1.0
Edge Energy Capacity	100 Joule
Fog Energy Capacity	2000 Joule
Energy Threshold	25%

B. Results

The Monte Carlo test is run 15 times, and each time it gives a different set of results. To find the mean, standard deviation, and 95% confidence interval for each metric, a t-distribution is used. Table 2 summarizes the Monte Carlo averaged performance of FCFS, EDF, and the proposed energy-aware priority scheduling framework.

TABLE 2
PERFORMANCE ANALYSIS OF SCHEDULING STRATEGIES

Strategy	Wait (s)	Viol (%)	SLA (%)	Energy (Joule)
FCFS	20.02	28.80	77.49	41876.2
EDF	9.33	12.10	91.10	21507.4
Proposed	8.77	11.50	91.62	15544.7

(i) Deadline Violation Analysis

The deadline violation analysis of the proposed approach is compared with the benchmark schemes in Figure 1. From the plot, it is observed that the proposed approach has lowest deadline violation ratio than EDF or FCFS approach since it takes into consideration energy awareness and task priority. This analysis shows that, beyond the task arrival pattern, processing tasks need to be done efficiently.

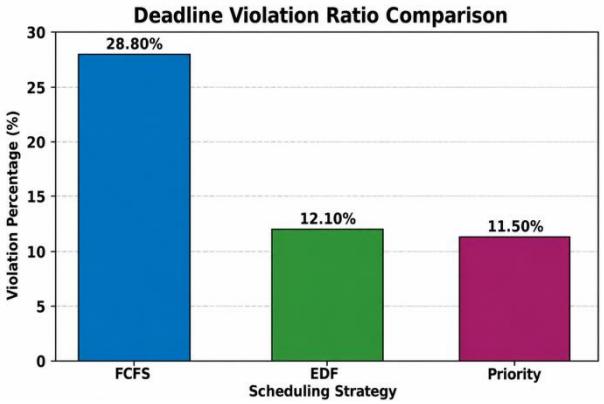

Fig 1: Deadline Violation Analysis

ISBN: 978-81-689089-0-1

(ii) Energy Consumption Analysis

The proposed scheme's energy consumption analysis is shown in Figure 2. The plot depicts that the proposed approach requires 27% and 63% less energy than EDF and FCFS approach, respectively. The major driver of this reduction is due to application of the energy-aware admission control and deactivation of underutilized nodes.

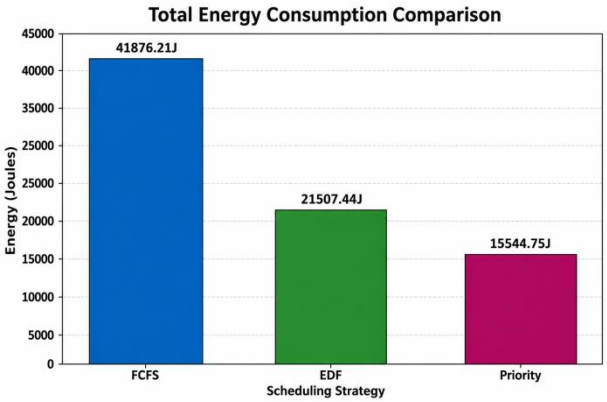

Fig 2: Total Energy Consumption Comparison

(iii) Critical Task SLA Satisfaction

The figure 3 shows the SLA satisfaction analysis of the proposed scheme for the most desired tasks compared with the other existing schemes. It is clear from the graph that the proposed approach gets the maximum level of SLA satisfaction as compared to the other approaches. This is due to the proposed scheme taking into account the task priority during processing and also introducing load balancing among resources like edge and fog nodes.

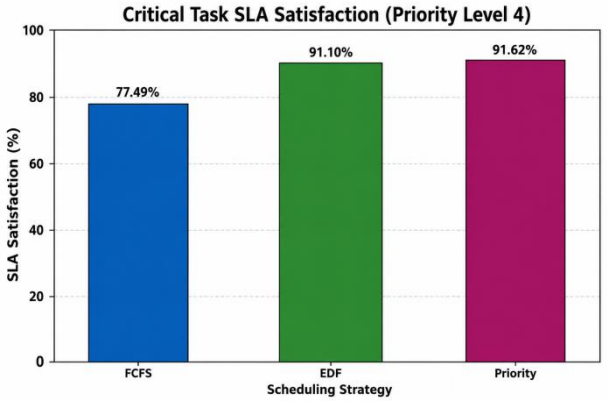

Fig 3: Critical Task SLA Satisfaction Comparison

(iv) *Waiting Time Analysis*

The proposed approach achieves the lowest waiting time compared to benchmark approaches, as depicted in Figure 4. Specifically, it exhibits approximately 60% lower waiting time than FCFS and about 6% lower than EDF. This improvement is attributed to the dynamic queueing mechanism, which manages the task queue efficiently

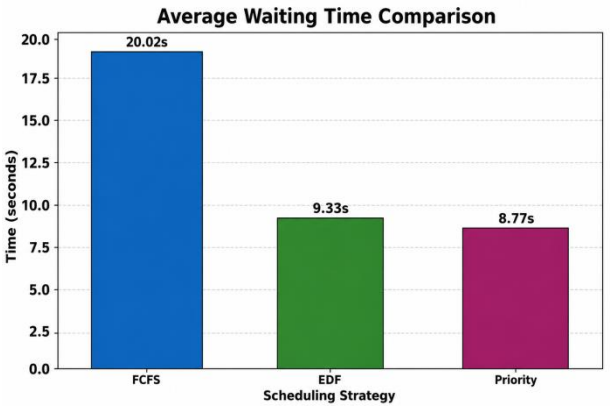

Fig 4: Waiting time Comparison

(v) Stress-Test Scenario (High Load)

The performance of the proposed approach is evaluated under a high-traffic scenario ($\rho \rightarrow 1$), where system stress is maximized. In this condition:

- Queue lengths grow significantly
- Deadline violations increase sharply in FCFS and EDF
- Edge nodes frequently enter into overloaded state
- Energy depletion accelerates

The proposed approach has fewer deadline violations and higher satisfaction with critical task SLAs because it uses selective priority handling and energy-aware admission control. This stress test demonstrates that the proposed framework remains more stable under near-saturation conditions.

C. Discussion

From the performance evaluation of the proposed approach and the existing approach, it is obvious that the energy-aware task scheduling with the priority concept makes the IoT system more stable under random workload conditions. The proposed solution also leads to reduced average waiting times and the number of missed deadlines, with a balanced resource usage at the edge nodes. The first thing to remember is that the suggested approach does increase the level of critical SLA satisfaction compared to the baseline approach. This highlights the need to ensure that task deadlines and priorities are included in the scheduling process. The analysis also shows the significance of the concept of smart offloading, cutting off the superfluous data offloading from the Cloud. This is also a significant aspect that contributes to the overall level of energy savings. The nodes are not exhausted at a too fast rate thanks to the ability to use the concept of queue-aware scoring and the level of residual energy. This renders the overall approach more trustworthy.

VI. CONCLUSION

This paper presents an energy-efficient and priority-based scheduling approach for IoT-Edge-Fog-Cloud environments .The approach combines queue length,

ISBN: 978-81-689089-0-1

CPU usage, residual energy, and priority level into a single framework. Simulation results show that the proposed approach improves waiting time, energy consumption, and the deadline violation rate compared to conventional task scheduling approaches. The statistical validation also demonstrates the robustness and reliability of the proposed approach..

REFERENCES

[1] Naaz, G. Joshi, and V. Sharma, "SAFED : secure and adaptive framework for edge - based data aggregation in IoT applications," Discov. Internet Things, 2025, doi: 10.1007/s43926-025-00143-3.

[2] Z. Naaz, G. Joshi, and V. Sharma, "Secure and Efficient Edge Computing Framework For IoT," pp. 6–10, 2023, doi: https://doi.org/10.1109/ICCCNT56998.2023.10307200.

[3] G. Joshi and V. Sharma, "A robust and trusted framework for IoT networks," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2022, doi: 10.1007/s12652-022-04403-w.

[4] Z. Naaz and N. Chauhan, "OTM : An Efficient Task Management and Offloading Scheme Using Proximity-Based Clustering , Fuzzy Logic , and Optimization in Edge-Cloud IoT Systems," pp. 1–29, 2026, doi: 10.1007/s10723-025-09820-7.

[5] M. S. Bali, K. Gupta, D. Gupta, G. Srivastava, S. Juneja, and A. Nauman, "An effective technique to schedule priority aware tasks to offload data on edge and cloud servers," Meas. Sensors, vol. 26, no. November 2022, p. 100670, 2023, doi: 10.1016/j.measen.2023.100670.

[6] C. Sonmez, A. Ozgovde, and C. Ersoy, "Fuzzy workload orchestration for edge computing," IEEE Trans. Netw. Serv. Manag., vol. 16, no. 2, pp. 769–782, 2019, doi: 10.1109/TNSM.2019.2901346.

[7] Z. Naaz and V. Sharma, "Performance Analysis of Edge Node in IoT Network," Proc. - Int. Conf. Comput. Power, Commun. Technol. IC2PCT 2024, vol. 5, pp. 546–551, 2024, doi: 10.1109/IC2PCT60090.2024.10486497.

[8] Q. Tang, B. Li, H. H. Yang, Y. Li, S. He, and K. Yang, "Delay and Load Fairness Optimization With Queuing Model in Multi-AAV Assisted MEC : A Deep Reinforcement Learning Approach," IEEE Trans. Netw. Serv. Manag., vol. 22, no. 2, pp. 1247–1258, 2025, doi: 10.1109/TNSM.2024.3520632.

[9] N. Chauhan and R. Agrawal, "A Probabilistic Deadline-aware Application Offloading in a Multi-Queueing Fog System: A Max Entropy Framework," J. Grid Comput., vol. 22, no. 1, 2024, doi: 10.1007/s10723-024-09753-7.

[10] D. Van Anh and V. Nguyen, "Leveraging Priority Queueing in IoT-Edge-Fog-Cloud Infrastructures for Efficient Healthcare Monitoring," IEEE Access, vol. 13, no. May, pp. 80461–80477, 2025, doi: 10.1109/ACCESS.2025.3565679.

[11] Z. Naaz, G. Joshi, and V. Sharma, "Load-balancing model using game theory in edge-based IoT network," Pervasive Mob. Comput., vol. 109, p. 102041, Apr. 2025, doi: 10.1016/j.pmcj.2025.102041.

[12] F. Saeik et al., "Task Offloading in Edge and Cloud Computing : A Survey on Mathematical , Artificial Intelligence and Control Theory Solutions," pp. 1–31.

[13] G. A. A. Blanco, E. N. R. Marcelli, A. R. S. Duarte, H.-T. Huang, K.-Y. Li, and T.-L. Chin, "Deadline-Aware Task Scheduling for Cloud-Fog Systems," in 2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN), IEEE, Mar. 2025, pp. 59–63. doi: 10.1109/ICIN64016.2025.10942963.

[14] Z. Naaz, G. Joshi, and V. Sharma, "SEDP: Secure and efficient data processing for heterogeneous IoT networks using punctured convolution coding," Eng. Res. Express, vol. 7, no. 2, p. 025266, Jun. 2025, doi: 10.1088/2631-8695/add85a.

[15] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, Oct. 2016, doi: 10.1109/TNET.2015.2487344.

Cite this article as:

Shaad, Vivek and et. al. "Energy-Aware Priority-Based Task Scheduling in IoT–Network", Proceedings of 13th international conference on Microelectronics, Circuits and Systems, Micro2026 (Vol-II)

Displayed as online on 25th July 2026.

Link: http://actsoft.org/science/micro2026-pro/350-micro2026.pdf

ISBN: 978-81-689089-0-1