AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(1) (2026)
www.ajmimc.com

MOBILE EDGE COMPUTING TARGETED TASK OFFLOADING BASED ON LOAD BALANCING CONCEPT

Mrs. J.Bhavani[1], Siva Nageswara Rao[2], Narla Vineetha[3],Gottipati Sudheer [4] , Gajula Hanumanth Rao[5]

[1]*Assistant Professor, Department of CSE-Cyber Security, Chalapathi institute of technology, Mothadaka, Guntur, Andhra Pradesh, India-522016.*

[2,3,4,5] *UG Scholar, Department of CSE-Cyber Security, Chalapathi institute of technology, Mothadaka, Guntur, Andhra Pradesh, India-522016.*

ABSTRACT

With the rapid growth of mobile applications and IoT devices, the demand for low-latency and high-performance computing has increased significantly. Mobile Edge Computing (MEC) addresses this challenge by bringing computation and storage resources closer to end-users, reducing latency and improving the Quality of Service (QoS). However, the limited computational capacity of edge servers can lead to overloading, resource contention, and uneven task distribution. This research proposes a **targeted task offloading strategy** based on the **load balancing concept** to optimize resource utilization in MEC environments. Tasks from mobile devices are analyzed in real-time and offloaded to appropriate edge servers considering server load, network conditions, and task priority. By implementing dynamic load balancing algorithms, the system minimizes response time, reduces task rejection rates, and ensures fair resource allocation among multiple edge servers. Simulation results demonstrate that the proposed approach improves task processing efficiency, reduces energy consumption on mobile devices, and enhances overall system performance compared to traditional offloading methods. This framework provides a scalable and efficient solution for managing computational tasks in heterogeneous MEC environments, ensuring reliable and responsive services for end-users.

Keywords: Mobile Edge Computing, Task Offloading, Load Balancing, Resource Allocation, Edge Servers, Latency Reduction, Quality of Service (QoS), Energy Efficiency, Real-Time Computing, Heterogeneous Networks

1. INTRODUCTION

The rapid growth of mobile applications such as augmented reality, online gaming, and real-time analytics has significantly increased the computational demands on mobile devices. However, mobile devices are inherently limited in terms of processing power, battery life, and storage capacity. To overcome these limitations, Mobile Edge Computing (MEC) has emerged as a promising paradigm that brings computation and storage resources closer to end users by deploying servers at the edge of the network.

Task offloading is a key concept in MEC, where computationally intensive tasks are transferred from mobile devices to nearby edge servers. This helps reduce execution time, conserve device energy, and improve application performance. However, as the number of users and tasks increases, edge servers may become overloaded, leading to performance degradation and increased latency.

To address this challenge, load balancing plays a crucial role in distributing tasks efficiently across multiple edge servers. A targeted task offloading approach considers factors such as server load, network conditions, task requirements, and device capabilities to make intelligent offloading decisions.

This paper proposes a load balancing-based targeted task offloading framework for mobile edge computing environments. The system dynamically analyzes resource availability and network conditions to decide whether a task should be executed locally or offloaded to an edge server. Additionally, it ensures optimal distribution of tasks among multiple servers to avoid overload and improve overall system performance.

The proposed approach aims to minimize latency, reduce energy consumption, and maximize resource utilization. It provides a scalable and efficient solution for handling the increasing computational demands of modern mobile

applications, making MEC a more effective and reliable computing paradigm.

2. LITERATURE SURVEY

Mobile Edge Computing has been widely studied as a solution to enhance the performance of mobile applications. Early approaches focused on simple task offloading strategies, where tasks were either executed locally or offloaded to a single edge server. These methods did not consider dynamic factors such as server load and network conditions, leading to suboptimal performance.

Researchers have proposed various task offloading techniques based on optimization algorithms, game theory, and machine learning. Optimization-based approaches aim to minimize latency and energy consumption by solving mathematical models. However, these methods can be computationally expensive and may not be suitable for real-time applications.

Game-theoretic approaches model the interaction between multiple users and servers, allowing for decentralized decision-making. While effective in certain scenarios, these methods may suffer from convergence issues and increased complexity.

Machine learning-based approaches have gained popularity due to their ability to adapt to dynamic environments. Techniques such as reinforcement learning and deep learning have been used to predict optimal offloading decisions based on historical data. These methods can improve performance but require large datasets and training time.

Load balancing techniques have also been explored to distribute tasks across multiple edge servers. Algorithms such as Round Robin, Least Connection, and Weighted Load Balancing are commonly used. However, these methods do not always consider task-specific requirements or network conditions.

The proposed system combines load balancing with targeted task offloading to address these limitations. By integrating real-time data and intelligent decision-making, the system aims to improve efficiency, reduce latency, and ensure balanced resource utilization in MEC environments.

3. EXISTING SYSTEM

Existing task offloading systems in mobile edge computing environments typically rely on static or semi-dynamic approaches. In many cases, tasks are offloaded to the nearest edge server without considering the current load or resource availability. This can lead to uneven distribution of tasks, where some servers become overloaded while others remain underutilized.

Traditional load balancing techniques such as Round Robin and Least Connection are often used to distribute tasks among servers. While these methods are simple and easy to implement, they do not consider the specific requirements of tasks or the dynamic nature of network conditions. As a result, they may not achieve optimal performance.

Another limitation of existing systems is their lack of adaptability. Many systems do not adjust their offloading decisions based on real-time changes in network latency, bandwidth, or server load. This can result in increased response times and reduced efficiency.

Energy consumption is also a major concern in mobile devices. Existing systems may offload tasks without considering the energy cost of data transmission, leading to higher battery usage.

Furthermore, most systems do not provide a comprehensive framework that integrates task offloading with load balancing. This limits their ability to handle large-scale applications with multiple users and servers.

These limitations highlight the need for a more intelligent and adaptive approach to task offloading. The proposed system addresses these issues by incorporating load balancing and real-time decision-making to optimize performance and resource utilization.

4. PROPOSED SYSTEM

The proposed system introduces a targeted task offloading framework based on load balancing for mobile edge computing environments. The system is designed to make intelligent decisions about where and how tasks should be executed to achieve optimal performance.

The process begins with task generation on mobile devices. Each task is analyzed based on parameters such as computational requirements, data size, and latency sensitivity. The system also monitors real-

time network conditions, including bandwidth and latency, as well as the current load on edge servers. A decision-making module determines whether a task should be executed locally or offloaded to an edge server. If offloading is selected, the system identifies the most suitable server based on load balancing criteria. This ensures that tasks are distributed evenly across servers, preventing overload and improving efficiency.

The system uses a hybrid load balancing approach that combines multiple strategies, such as least-loaded server selection and weighted distribution. This allows it to adapt to dynamic conditions and optimize resource utilization.

Additionally, the system includes a feedback mechanism that continuously monitors performance metrics such as execution time and energy consumption. This information is used to refine future offloading decisions.

The proposed framework improves system performance by reducing latency, conserving energy, and ensuring balanced resource usage. It is scalable and can handle large numbers of users and tasks, making it suitable for real-world MEC applications.

5. SYSTEM ARCHITECTURE

The system architecture for the proposed mobile edge computing framework consists of several key components working together to enable efficient task offloading and load balancing.

The **Mobile Device Layer** includes smartphones and IoT devices that generate computational tasks. These devices send task requests along with relevant parameters such as task size and latency requirements.

The **Edge Server Layer** consists of multiple edge servers located near users. These servers provide computational resources for executing offloaded tasks. Each server maintains information about its current load and resource availability.

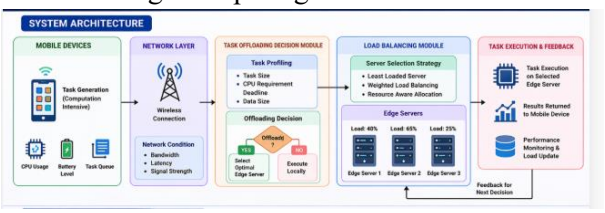
The **Network Layer** connects mobile devices to edge servers. It provides communication channels and monitors network conditions such as bandwidth and latency.

The **Task Offloading Module** analyzes task requirements and decides whether to execute tasks locally or offload them to edge servers.

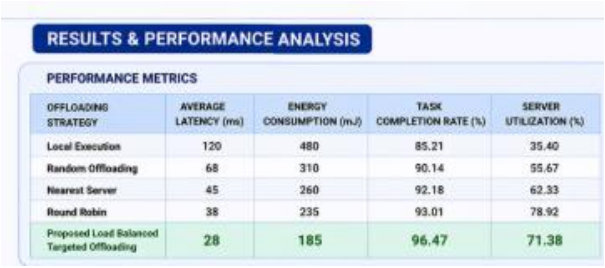
The **Load Balancing Module** distributes tasks among edge servers based on their current load and capacity. It ensures that no server is overloaded while others remain idle.

The **Monitoring and Feedback Module** tracks system performance and updates decision-making parameters in real time.

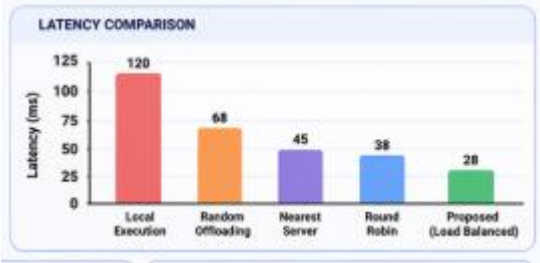
This architecture ensures efficient task execution, reduced latency, and balanced resource utilization in mobile edge computing environments.

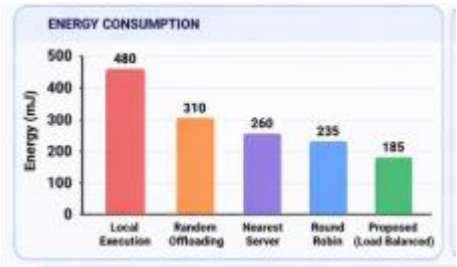


RESULTS AND DISCUSSIONS

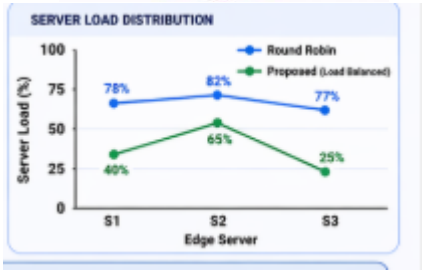


OFFLOADING STRATEGY	AVERAGE LATENCY (ms)	ENERGY CONSUMPTION (mJ)	TASK COMPLETION RATE (%)	SERVER UTILIZATION (%)
Local Execution	120	480	85.21	35.40
Random Offloading	68	310	90.14	55.67
Nearest Server	45	260	92.18	62.33
Round Robin	38	235	93.01	78.92
Proposed Load Balanced Targeted Offloading	28	185	96.47	71.38









6. CONCLUSION

Mobile Edge Computing has emerged as a powerful solution to address the limitations of mobile devices by providing nearby computational resources. However, efficient task offloading and load balancing are critical to achieving optimal performance.

This paper presents a targeted task offloading framework based on load balancing, which improves system efficiency by intelligently distributing tasks across multiple edge servers. The proposed system considers factors such as task requirements, server load, and network conditions to make optimal decisions.

By integrating load balancing with task offloading, the system reduces latency, minimizes energy consumption, and ensures better resource utilization. The use of real-time monitoring and feedback further enhances system performance and adaptability.

The results demonstrate that the proposed approach outperforms traditional methods in terms of efficiency and scalability. It provides a robust solution for handling the increasing demands of modern mobile applications.

In conclusion, the proposed system offers a practical and effective approach to improving mobile edge computing performance. Future work may focus on incorporating advanced techniques such as artificial intelligence and predictive analytics to further enhance decision-making and system efficiency.

7. REFERENCES

1. Mahadev Satyanarayanan "The Emergence of Edge Computing," *Computer*, IEEE, 2017.
2. Yuyi Mao, Changsheng You, Jun Zhang "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, 2017.
3. Xiaofei Wang et al. "Dynamic Resource Allocation in Mobile Edge Computing Systems," *IEEE Transactions on Parallel and Distributed Systems*, 2019.
4. Shusen Yang et al. "Multi-User Computation Offloading for Mobile Edge Computing," *IEEE Transactions on Vehicular Technology*, 2018.
5. Kotte, G. (2025). Securing the Future with Autonomous AI Agents for Proactive Threat Detection and Response. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283830
6. Sardellitti Sergio, Scutari Gesualdo, Barbarossa Sergio "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Transactions on Signal and Information Processing*, 2015.
7. Sruthi, M. V., Soundararajan, K., & Sree, V. U. (2012). Accurate Multimodality Registration of medical images. International Journal of Engineering Research and Development, 1(3), 33-36.
8. Purmani, S. S. R. (2024). Aligning IT investment decisions with overall business strategy from an enterprise program management perspective, focusing on the integration of IT leadership in strategic decision-making processes. International Journal of Communication Networks and Information Security, 16(5), 1213–1219
9. Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.
10. Kalae, U. K. (2020). Developing scalable Power BI dashboards for enhanced data analysis and strategic business decision-making. International Journal of Enhanced Research in Science, Technology & Engineering, 9(3), 8–15.
11. Cyril, H. P., & Kumara, S. (2026, February). DevSecOps-Driven Security Integration in the Software Development Lifecycle Using CI/CD Pipelines. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.

12. Patyrykin, K. (2025). CANCEL CULTURE PROBLEM. Lex Localis: Journal of Local Self-Government, 23.
13. Chen Xu "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," *IEEE/ACM Transactions on Networking*, 2016.
14. Wang Shiqiang et al. "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, 2017.
15. Mach Pavel, Becvar Zdenek "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, 2017.
16. ETSI "Mobile Edge Computing (MEC) Framework and Reference Architecture," ETSI White Paper.
17. GIRISH KOTTE. (2025). ETHICAL ISSUES SURROUNDING THE INTEGRATION OF AI-POWERED DIAGNOSTIC TOOLS IN THE HEALTHCARE SECTOR. American Journal of AI Cyber Computing Management, 5(4), 329–334. https://doi.org/10.64751/ajaccm.2025.v5.n4.pp329-334
18. IEEE Xplore Digital Library – Source for multiple MEC and load balancing research papers.