Iraqi Journal of Information and Communications Technology(**IJICT**)
Vol. 7, Issue 3, December 2024
www.ijict.edu.iq
ISSN:2222-758X
e-ISSN: 2789-7362

SIMPLIFIED DISTRIBUTED LEDGER FOR TASK OFFLOADING IN EDGE NETWORKS

Sarah R. Al-Hafidh [1], **Emad H. Al-Hemiary** [2]
[1,2] Department of Networks Engineering, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq
sarah.rabea@yahoo.com[1], emad.alhemiari@gmail.com[2]
Corresponding Author: **Emad H. Al-Hemiary**
Received:11/05/2023; Revised:22/10/2023; Accepted:03/12/2023
DOI:**10.31987/ijict.7.3.247**

***Abstract*- Large quantities of processing resources with strict latency specifications are needed for Internet of Things (IoT) devices due to the rise of compute-intensive and delay-sensitive mobile apps. One promising solution is to transfer resource-intensive computational tasks from IoT devices to either edge computing servers or cloud computing servers. This paper aims to apply a simplified distributed ledger to an edge network to follow up the offloaded data and maintain the response time as much as possible. The voting process is used as a consensus to validate the new block, while the offloading decision is based on a fixed processing time offloading threshold value. The proposed model has been programmed and the experimental evaluation of the proposed model shows that the ledger did not significantly lengthen the response time and the offloaded task has been successfully tracked.**

keywords: **Edge offloading, IoT offloading, Consensus, Blockchain, Data tracking.**

I. INTRODUCTION

Edge-Cloud platforms that facilitate IoT applications usually consist of 3- tiers: the end device tier, the edge tier, and the Cloud (Datacenter) tier. The edge layer comprises multiple geographically distributed edge nodes, which are connected to the Cloud tier through the core network. Both the edge nodes and end user devices are directly connected to each other [1]. Edge and cloud computing play a crucial role in various applications, including: healthcare facilities, Augmented Reality (AR), traffic management systems, enhanced networking through caching and processing, and the development of smart homes and cities. These technologies are extensively utilized to support and enhance the capabilities of these diverse applications [2,3].

In addition, offloading refers to the process of transferring computational tasks from resource-constrained end devices to resource-rich nodes. This approach is employed to enhance application performance and improve energy efficiency. By offloading computations to nodes with more resources, the burden on the end device is reduced, resulting in better overall performance and more efficient energy utilization [2]. Offloading in an IoT network might take place from a cloud node; from one edge node to another edge node, or from an end device to an edge node. Offloading decision-making can be carried out locally or globally. In the local system, the decision could be made by taking into account how mobile users perceive the situation. In the global scheme, the choice could be made taking into account the state of the entire system. The offloading choice in this plan can be carried out through three ways [4]: reactive, proactive, or hybrid. However, the main advantages of offloading are: reduce response time, reduce power consumption, improve performance, and support the new applications that may require resources-rich nod for their computations and storage [5]. The response time refers to the time interval between data collection for decision-making and the presentation of the outcome to the customer [6].

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN: 2789-7362

During the process of task offloading, there is a significant possibility of encountering data loss or privacy breaches [7]. Since blockchain technology is distributed and has unique security can be used as a potential solution for ensuring data security and data tracking. Blockchain is defined as a data structure that uses hash functions and asymmetric encryption methods to prevent data fraud and manipulation. The most common hash function used in blockchain is SHA256, which is defined as a type of Secure Hash Algorithm (SHA), and it produces an output of 256 bits, which is the same as having a 32-byte array. When represented in hexadecimal form, this results in a string of 64 characters [8]. Blockchainâs data blocks are chronologically arranged representations of user transfers [9]. In a blockchain network, every newly created block undergoes validation by all participating nodes. Once validated, the block is appended to the end of the blockchain. This process known as consensus, which ensures an agreement among the nodes. Common consensus algorithms are used such as, Proof of Work (PoW), Proof of Stake (PoS), and practical Byzantine fault tolerance (PBFT) [10]. There are many blockchain applications for the IoT networks such as, Internet of Healthcare Things (IoHT), Internet of Vehicles (IoV), Internet of Energy (IoE), IoT devices, and many other applications [11]. Blockchain is considered a foundational technology in the context of 6G, as it offers numerous advantages including the ability to trace extensive amounts of data, enable autonomous interactions among various IoT systems, promote interoperability across devices, and enhance the reliability of 6G communication systems with massive connectivity [12].

This study focuses on the design and implementation of an edge offloading network that incorporates a simplified distributed ledger. The objective is to streamline the consensus process of blockchain while minimizing the impact on task response time and ensuring a balanced load distribution between edge nodes and the cloud. The remaining sections of the paper are organized as follows: Section II presents an overview of related works, Section III describes the proposed model, Section IV provides details on the implementation, Section V represents the simulation results, Section VI discusses the obtained results, and finally, Section VII concludes the study.

II. LITERATURE SURVEY

The combination of blockchain's privacy protection, immutability, and traceability features with edge computing can enhance the security of edge computing in IoT applications. Additionally, edge computing can provide ample resources to ensure the high-performance functionality of blockchain systems [13].

To address the objective of minimizing network utility, which includes both the energy consumption of the fog computing network (FCN) and the latency of the blockchain network. The authors in [14], Xiaoge Huang and his colleagues, introduced a fog computing network integrated with blockchain. The network architecture in [14] consists of two layers: the device layer and the fog layer, with the blockchain technology deployed in the fog layer.

In [7], the authors, Huaming Wu, Member, IEEE, Katinka Wolter, et al., introduced the Energy Efficient Dynamic Task Offloading (EEDTO) algorithm to address the twin goals of minimizing energy usage and reducing task response times. This algorithm makes real-time decisions on the most suitable computing location, which could be either the IoT device, the Mobile Edge Computing (MEC) server, or the Mobile Cloud Computing (MCC) server. The control of costs related to computation and communication was managed using the Lyapunov optimization technique. Furthermore, the paper proposes

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN: 2789-7362

the incorporation of a simplified blockchain structure into edge offloading, facilitating the tracking of tasks from IoT devices to the Edge or Cloud environments.

In [15], the authors, Xiaolong Xu et al., presented the BeCome (blockchain-enabled computation offloading method) approach to tackle several objectives simultaneously. Its primary focus is on reducing task offloading time and energy consumption for edge computing devices, all while maintaining load balancing and data integrity. By harnessing the power of blockchain technology, the BeCome approach offers an effective and efficient framework to achieve these objectives.

In [16], the authors, Amira S. Ibrahim, Hassan Al-Mahdi, and Hamed Nassar, used the queueing theory to develop a queueing model, where decisions on offloading are dependent on the required processing time for each task. A specific, predetermined processing time threshold regulates offloading.

In [17], the Reputation and Voting based Consensus (RVC) algorithm introduced as a blockchain consensus mechanism specifically designed for edge computing-enabled IoT systems. The RVC algorithm aims to achieve a secure and efficient consensus mechanism by incorporating a reputation calculation method. It also focuses on enabling high transaction throughput to meet the demands of edge computing environments. The RVC algorithm provides a reliable and robust consensus mechanism for edge computing-enabled IoT systems.

In [18], the authors, Ola M. Al-Tuhafi and Emad H. Al-Hemiary, proposed an adaptive offloading algorithm that dynamically adjusts and identifies the optimal offloading threshold value. This approach ensures a balanced workload between the edge nodes and resource-constrained end devices, even when the load fluctuates. By dynamically adapting to changes in the load, the algorithm aims to maintain a consistent level of resource allocation and offloading, thereby optimizing the performance of the system.

III. PROPOSED MODEL

The proposed model involves N terminal devices and M edge nodes. Each terminal device generates tasks at random intervals, with the time it takes to process a task following an exponential distribution with an average denoted as μ. The decision to offload a task to an edge node is based on a processing time threshold, τ. This threshold determines whether a task gets offloaded. As a result, the time it takes for an edge node to provide service follows a truncated exponential distribution, which gives us a distribution that's more general than a purely exponential one. This essentially means that the queuing system for each terminal device is characterized as M/G/1, while the queuing systems for each edge node are represented as M/G/m. As illustrated in Fig. 1, the first region, denoted as α, represents the probability of processing the task directly at the end device. The second region β is the probability of processing the task at edge node. α and β can be obtained as in[19,20] as follows:

$$\alpha = \mu \int_0^\tau e^{-\mu t}\, dt = 1 - e^{-\mu\tau} \tag{1}$$

$$\beta = \mu \int_\tau^\infty e^{-\mu t}\, dt = e^{-\mu\tau} \tag{2}$$

ISSN:2222-758X
e-ISSN: 2789-7362

The arrival rate to the terminal device CPU is represented as λ_{TD}, and the arrival rate to the edge node is denoted as λ_{EN}. Both λ_{TD} and λ_{EN} are assumed to follow Poisson processes. These arrival rates can be obtained as [19,20] as follows:

$$\lambda_{\text{TD}} = \alpha\lambda \tag{3}$$

$$\lambda_{\text{EN}} = N\beta\lambda \tag{4}$$

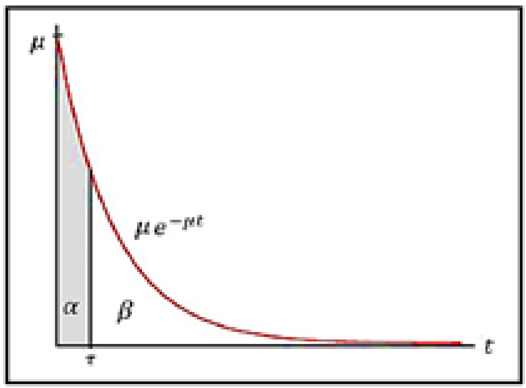


Figure 1: The offloading threshold τ partitions the exponential distribution of service time into two truncated exponentials [18].

The average response time of the M/G/1 queue can be calculated using Pollaczek-Khinchine formula for mean values, and is expressed as:

$$W = \mathbb{E}[s] + \frac{\lambda\mathbb{E}[s^2]}{2(1-\lambda\mathbb{E}[s])} \tag{5}$$

where W is the mean response time, $\mathbb{E}[s]$ is the first moment, and $\mathbb{E}[s^2]$ is the second moment of the service time distribution.

The average response time of the M/G/1 queue can also be approximated using an extended Pollaczek-Khinchine formula for mean values, expressed as in [21]:

$$W \approx \mathbb{E}[s] + \frac{\lambda^n\mathbb{E}[s^2](\mathbb{E}[s])^{n-1}}{2(n-1)!(n-\lambda\mathbb{E}[s])^2 G} \tag{6}$$

where G is given by:

$$G = \sum_{i=0}^{n-1} \frac{(\lambda\mathbb{E}[s])^i}{i!} + \frac{(\lambda\mathbb{E}[s])^n}{(n-1)!(n-\lambda\mathbb{E}[s])} \tag{7}$$

W is the mean response time; n is the number of identical parallel servers in the system, $\mathbb{E}[s]$ is the first moment and $\mathbb{E}[s^2]$ is the second moment of the service time distribution.

Once the decision is made to offload the task to the edge, end device will create a block for this task. The block consists of header and body, the header contains the following information: terminal device ID (TD_ID), timestamp, previous hash, and current hash, and the body of the block will contain the task as shown in Fig. 2.

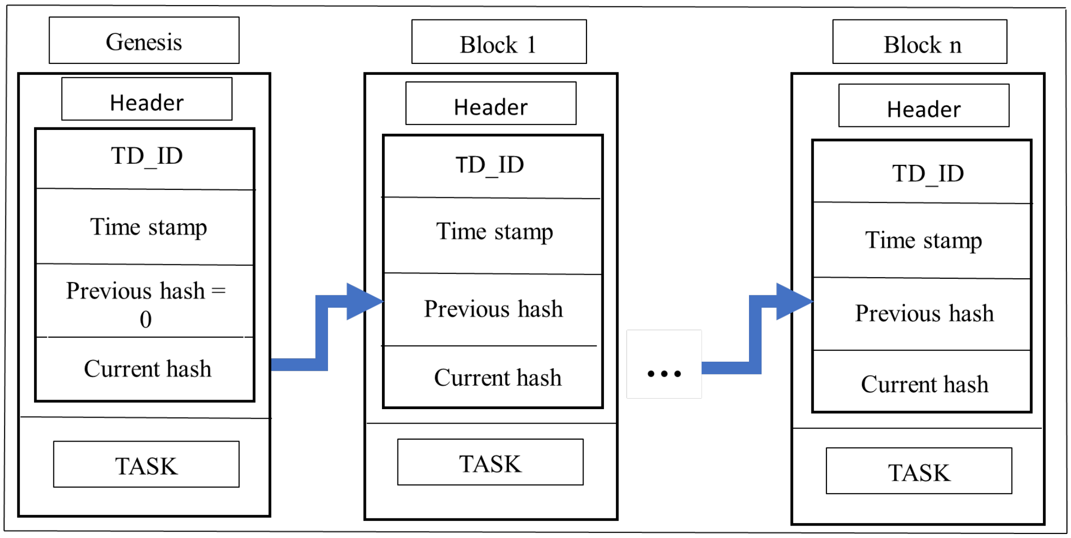


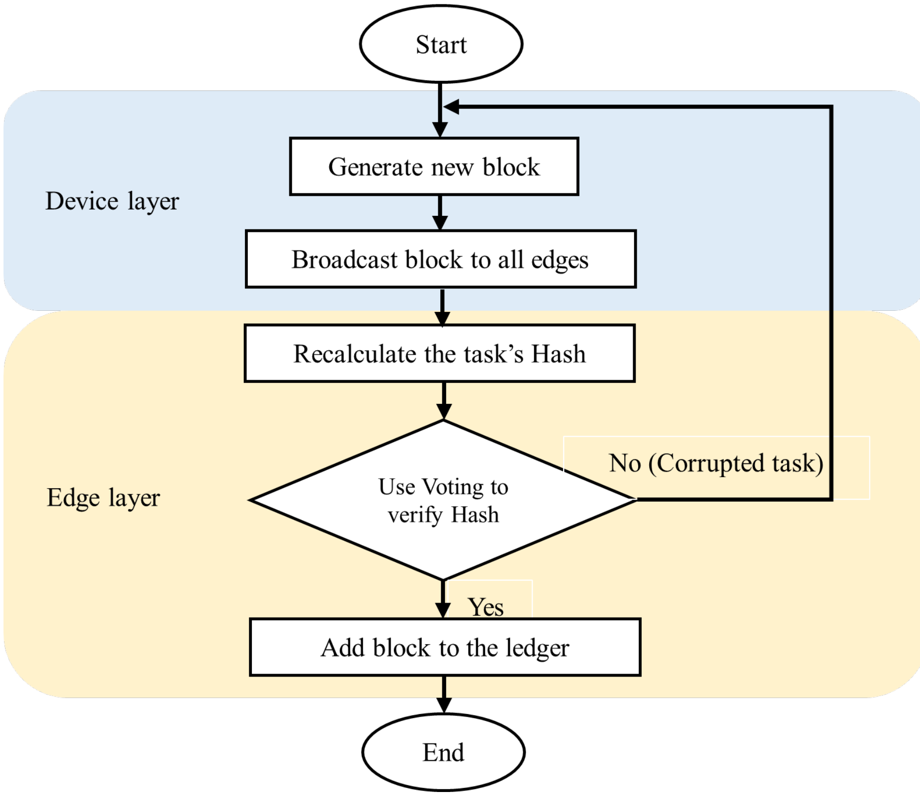
Figure 2: The proposed simplified distributed ledger data structure.

Through the (TD_ID) and time stamp which refer to the time of creating the task as well as creating the block, the traceability of the task will be achieved, and the hash will create the blockchain. The block will be broadcast to all edge nodes, each edge node will recompute the hash (SHA256) of the task and compare it with the current hash that stored in the received block, if they are equal, it will vote yes and if they are not, it will vote no. The votes will be shared with all edge nodes. If the number of yes votes > no votes the block will be appended to the end of the ledger that stored in each edge node as shown in Fig. 3 (a) and (b).

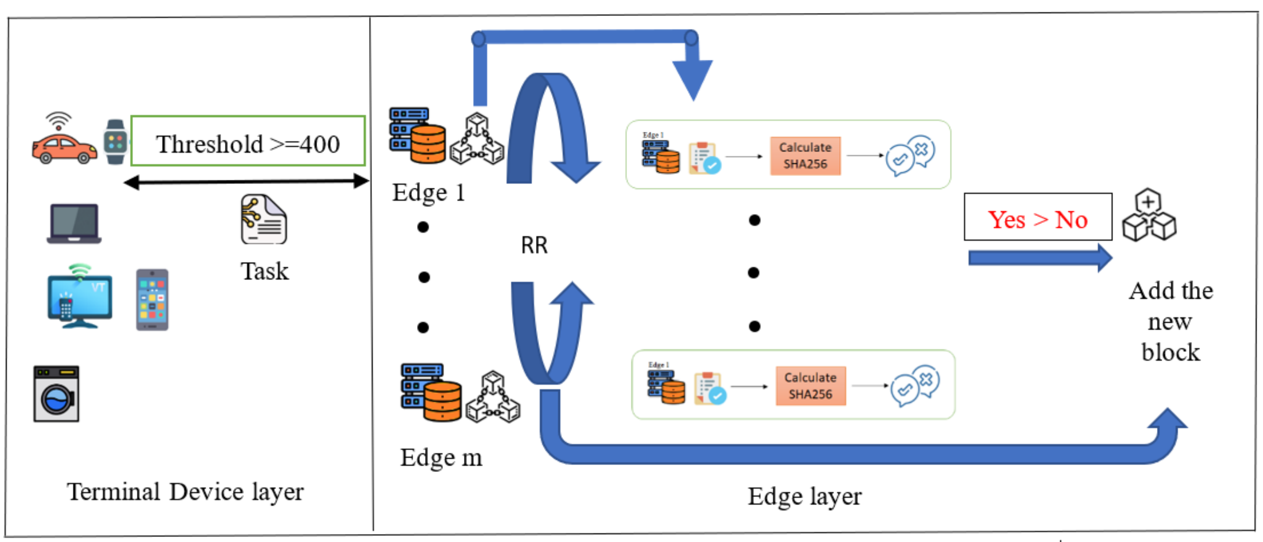
IV. IMPLEMENTATION OF PROPOSED MODEL

The suggested model was put into action on a local laptop, and the simulation of the model was performed using the C++ programming language. The work environment chosen was the most recent version of Microsoft Integrated Development Environment (IDE), specifically Microsoft Visual Studio 2022, Version 17.4. Multiple experiments were conducted to showcase the system's efficiency. Each outcome of these experiments was meticulously logged and cross-verified following simulations that spanned over a period of 34 real-time days, which is equivalent to 3 million seconds. This extended duration was essential for gathering an ample amount of task data and obtaining an accurate measure of the average response time. In the proposed network, there is $N = 400$ end device, the queueing model of each end device is assumed to be $M/G/1$ with a CPU processing rate $\mu = 0.001$ task per second. All end devices are offloading their extensive computation task to corresponding edge nodes, the number of edge nodes is $M = 5$ and the queueing model at the edge layer is $M/G/k$. and

Iraqi Journal of Information and Communications Technology(IJICT)
Vol. 7, Issue 3, December 2024
www.ijict.edu.iq
ISSN:2222-758X
e-ISSN: 2789-7362



(a)



(b)

Figure 3: (a) Flowchart of the blockchain consensus, and (b) System model of simplified distributed ledger for task offloading in Edge networks.

ISSN:2222-758X
e-ISSN: 2789-7362

each edge node accelerates processing by 20 times as compared to the end device's processing rate; i.e., $X = 20$.

V. RESULTS

A. Experiment (A):

The network was examined before applying blockchain on it, in order to evaluate the effect of adding blockchain technology on the mean response time. The offloading model was programmed with a fixed threshold $\tau = 400$, this threshold has been chosen after numerous experiments to balance resource consumption. While the arrival rate λ varies from 0.00001 to 0.00019 task per second. The result in Fig. 4, shows the response time of terminal devices, and the response time of edge nodes without applying blockchain.

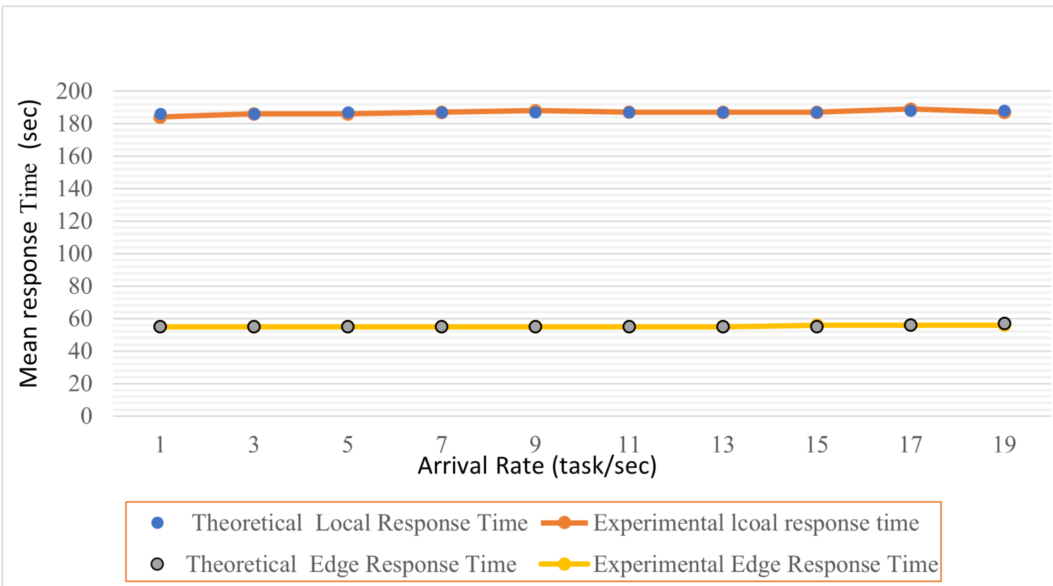


Figure 4: Result of offloading algorithm.

B. Experiment (B):

To evaluate the effectiveness of the proposed network, the same parameters of the previous network has been used, while applying the distributed ledger to the offloaded tasks, ranging in size from 100KB to 1MB. The impact of incorporating the ledger into the offloading system, with a fixed offloading threshold and a task size range between 100KB to 1MB, was thereby demonstrated. The result in Fig. 5, shows the required time for validating the new block and appending it to the end of the ledger, note that the time of sharing the vote among edges is negligible. While Fig.6 shows the response time after applying the distributed ledger.

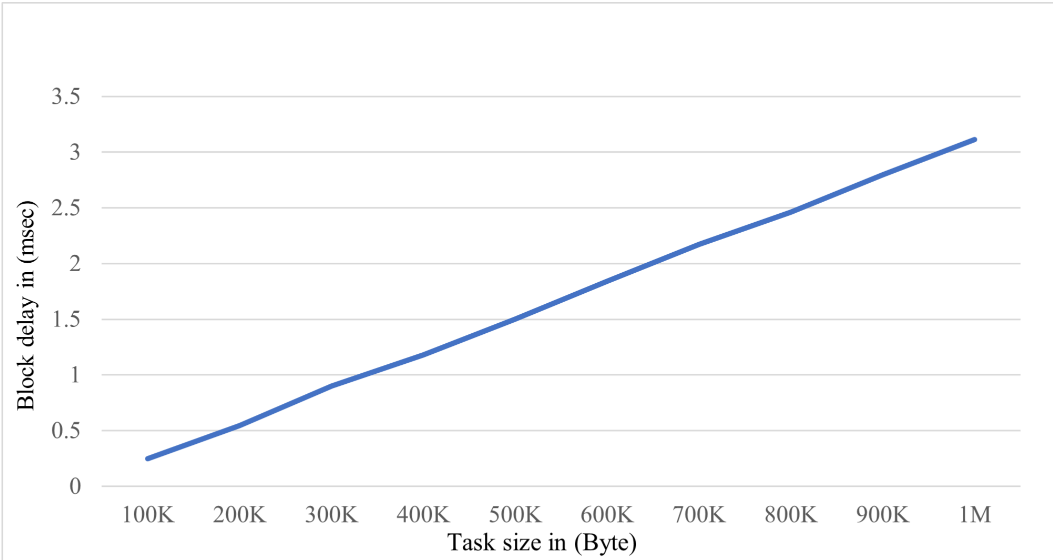


Figure 5: Block delay of the distributed ledger.

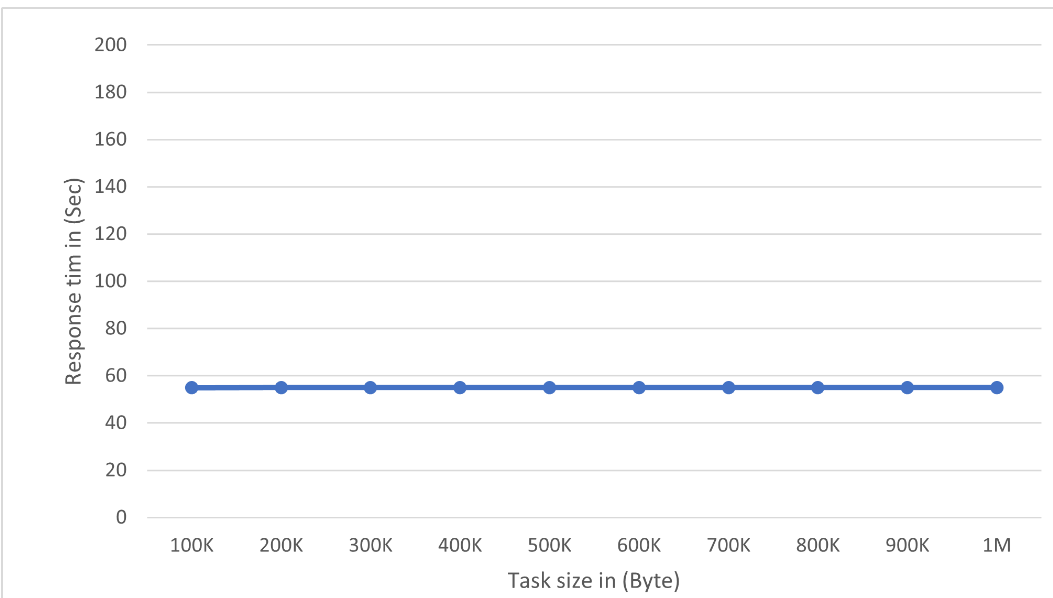


Figure 6: Response time with blockchain.

C. Experiment (C):

In order to evaluate the proposed model effectiveness, it has been compared to the RVC method. The number of edge nodes has been increased from (50-300) nodes and the arrival rate (λ) were equal to 0.00001. Fig. 7 shows the comparison between RVC and the proposed model of the paper.

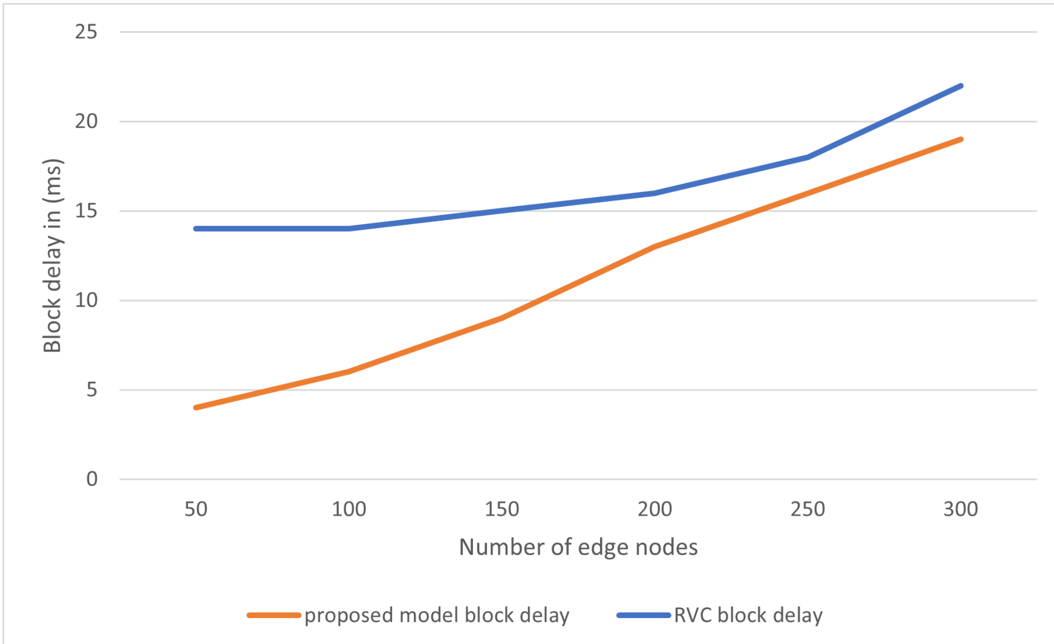


Figure 7: Comparison between RVC and the proposed model.

VI. DISCUSSION

The obtained results reveal significant differences in the processing times between offloaded tasks and terminal device tasks. Specifically, the offloaded tasks, with a processing time greater than 400, exhibited a processing time around 55 seconds, whereas the terminal device tasks, with a processing time less than 400, took around 180 seconds. This indicates that offloading intensive tasks to the edge node effectively reduces the processing time, resulting in improved performance compared to local processing on the terminal device. Moreover, the application of a blockchain in the system is notable. The calculated delay of the consensus process was found to be very small, measured in milliseconds, and did not have a noticeable impact on the response time of the offloaded tasks. This indicates that the blockchain implementation was efficient, as it maintained the integrity and authenticity of the system without significantly affecting the overall performance. Furthermore, comparing the proposed model to the RVC, the time required to validate the new block and added to the ledger in the distributed ledger was less than in RVC. However, it's worth noting that the RVC algorithm offers greater security as it utilizes a combination of Merkle trees and reputation-based voting, which makes it more secure than the proposed distributed ledger model. It is worth noting that the response time has been calculated theoretically according to eqs. (5) and (6), and experimentally.

VII. CONCLUSIONS

The proposed model is implemented by applying blockchain technology to an offloading algorithm. The offloading threshold was kept fixed and the blockchain consensus was simplified. The result evaluations showed that it is possible to use blockchain with offloading without increasing response time significantly. With task size ranging from (100KB-1MB) the increasing of response time is almost non- existent. It is worth noting that if we kept increasing the size of the task, the time required for the new block will also increase and the model may breakdown at a specific task size.

Funding

None

ACKNOWLEDGEMENT

The author would like to thank the reviewers for their valuable contribution in the publication of this paper.

CONFLICTS OF INTEREST

The author declares no conflict of interest

REFERENCES

[1] J. Almutairi , M. ALdossary, "Modeling and Analyzing Offloading Strategies of IoT Applications over Edge Computing and Joint Clouds," symmetry, vol. 13, p. 402, march 2021.
[2] K. Gasmi, S. Dilek, S. Tosun and S. Ozdemir, "A survey on computation offloading and service placement in fog computing-based IoT," The Journal of Supercomputing volume, vol. 78, pp. 1983-2014, june 2021.
[3] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, âToward Computation Offloading in Edge Computing: A Survey,â IEEE Access, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 131543â131558, 2019
[4] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh,, "A Survey on the Computation Offloading Approaches in Mobile Edge/Cloud Computing Environment: A Stochastic-based Perspective,," Journal of Grid Computing, vol. 18, no. 4, p. 639â671, 2020.
[5] M. Masdari and H. Khezri, âEfficient offloading schemes using Markovian models: a literature review,â Computing, vol. 102, no. 7, pp. 1673â1716, Jul. 2020
[6] S. Shahhosseini, A. Anzanpour, I. Azimi, S. Labbaf, D. Seo, S.S. Lim, P. Liljeberg, N. Dutt and A. M. Rahmani, "Exploring computation offloading in IoT systems," Journal of Information System, vol. 107, p. 101860, 2022.
[7] H. Wu, K. Wolter, P. Jiao, Y. Deng, Y. Zhao and M. Xu, "EEDTO: An Energy-Efficient Dynamic Task Offloading Algorithm for Blockchain-Enabled IoT-Edge-Cloud Orchestrated Computing," in IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2163-2176, Feb.15, 2021.
[8] H. Huang, J. Tian, G. Min, W. Miao. (2020). Introduction to blockchains. In L. W. H. Huang, Blockchains for Network Security (pp. 10-13). London, United Kingdom: The Institution of Engineering and Technology.
[9] X. Fu, H. Wang, and P. Shi, "A survey of Blockchain consensus algorithms: mechanism, design and applications," Science China Information Sciences, vol. 64, p. 121101, 2021.
[10] H. -N. Dai, Z. Zheng and Y. Zhang, "Blockchain for Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076-8094, Oct. 2019.
[11] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras and H. Janicke, "Blockchain Technologies for the Internet of Things: Research Issues and Challenges," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2188-2204, April 2019.
[12] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," in IEEE Open Journal of the Communications Society, vol. 1, pp. 957-975, 2020.
[13] H. Xue, D. Chen, N. Zhang, H.-N. Dai, and K. Yu, "Integration of blockchain and edge computing in internet of things: A survey," Future Generation Computer Systems, vol. 144, pp. 307-326, 2023
[14] X. Huang, X. Deng, C. Liang, and W. Fan, "Blockchain-Enabled Task Offloading and Resource Allocation in Fog Computing Networks," Wireless Communications and Mobile Computing, vol. 2021, p. 12, 2021.
[15] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing," IEEE Transactions on Industrial Informatics, vol. 16, no. 6, pp. 4187-4195, June 2020.
[16] A. S. Ibrahim, H. Al-Mahdi, and H. Nassar, "Characterization of task response time in a fog-enabled IoT network using queueing models with general service times," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 7089-7100, 2022
[17] Z. Liao and S. Cheng, "RVC: A reputation and voting based blockchain consensus mechanism for edge computing-enabled IoT systems," Journal of Network and Computer Applications, vol. 209, p. 103510, 2023.
[18] O. M. Al-Tuhafi and E. H. Al-Hemiary, "Adaptive Thresholds for Task Offloading in IoT-Edge-Cloud Networks," 2023 International Conference on Cyber Management and Engineering (CyMaEn), Bangkok, Thailand, 2023, pp. 379-385.
[19] V. I.-T. U. of Denmark and undefined 2010, âTeletraffic engineering and network planning,â unina.stidue.net, 2009, Accessed: Sep. 07, 2022.

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN: 2789-7362

[20] I. Mohamed, H. Al Mahdi, M. Tahoun, H. Nassar, "Characterization of task response time in fog enabled networks using queueing theory under diferent virtualization modes," Journal of Cloud Computing: Advances, Systems and Applications, vol. 11, p. 21, 2022.

[21] W. J. Stewart, Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling, 2009.

