International Journal of Electrical Engineering & Emerging Technology, Vol. 06, No. 01, JUN 2023, pp 13-19

A Delay-Aware Energy-Efficient System for a 5G MEC/D2D Network Using a Computational Selection Algorithm

[a] Muzamil Hussain
[a]School of Computer Beijing University of Posts and Telecommunications, China
Corresponding author e-mail: (mlhmemon2@gmail.com)
[**Received on:** 16/05/2023 **Accepted on**: 08/06/2023 **Published on:** 21/06/2023]

Abstract — **The challenge now is to find a way to combine D2D and MEC into a low latency, energy-efficient system. The main goal of this study is to present a delay-aware, energy-efficient system for carrying the D2D capabilities of the MEC network. As these two technologies are built into 5G. To evaluate the effectiveness implemented a simulation. Also show that the proposed EEDOS The energy saving rate is shown in simulation iterations, but the number of units is limited to 100 nodes. To evaluate the effect of work on energy savings, the probability of task completion was raised in this study from 30% to 70%. Increasing the task allocation rate in the latter reduced the number of units available in the neighborhood. EEDOS, on the other hand, consistently saves energy in the range of 95.8% to 96.9% in both cases. The D2D mitigation is 50% more energy efficient than when working on local hardware. The average energy efficiency of their system is 60%. EEDOS saved 97% of energy compared to local data execution. The results show that EEDOS outperforms existing data relief solutions in terms of energy efficiency and data execution time for mobile / IoT devices when compared to edge computing data relief and D2D collaboration. EEDOS requires significantly less server computer capacity than the MEC unloading schedule. Finally, EEDOS reduced the number of missed deadlines compared to previous research.**

Index Terms — **Mobile Edge Computing (MEC) Technology, Energy Minimization, Optimization Algorithms.**

I. INTRODUCTION

THE mobile edge computing is an advance field and provides many of the computation and energy related solutions for the devices. Many of the solutions have been provided so that the efficiency and accuracy can be provided in a condition that the energy can be saved, and the computation power should be supported from the edge of the network placed on the base stations and access points. One of the solutions is the offloading so that the energy can be saved of the mobile devices and latency can be minimized. The main goal of this study is to speed up task completion while improving the energy efficiency of mobile devices and the Internet of Things (IoT). The author first recognized the problem and gave an overview of his research objectives in this chapter. The author then explained why this study was important and how the study was conducted. Finally, in the last part, I explained the structure of the dissertation. The use of smartphones has become commonplace, requiring users to use their mobile phones for extended periods of time during the day. However, mobility and design limitations prevented it from meeting today's resource-intensive application tasks. Applications that require high computer power and low latency networks include augmented reality, facial recognition, and online games [1]. MEC and D2D communication are two 5G technologies that save energy and eliminate communication and computing delays between mobile and IoT devices. MEC is presented as a solution to the long latency of cloud computing when releasing data. The MEC design reduces the need for time-consuming data transfers over the main network. Instead of transferring data to the cloud, MEC receives data from end users or IoT devices and processes it at the edge of the network. The energy of these devices is saved with a slight delay after downloading the computationally intensive data to MEC. In addition, you can use data relief to support real-time applications [2]. As a proximity service, D2D communication was originally provided by the Cellular Network (ProSe) LTE Tel-12 specification. This is a promising 5G technology that allows low-capacity devices to get service from more powerful neighbors without interference from base stations. The number of devices connected to a single 5G cell increase. It is more likely that many powerful devices will appear near resource-limited devices. As a result, task relief can be fully utilized through D2D collaboration. This technique can be used by devices with limited resources, such as CPU and power, to transfer work to higher performance devices, conserving power and reducing processing delays. Note that the focus of this research was on the benefits of working with local hardware for IoT and mobile devices [3]. Augmented Reality (AR) is emerging as one of the technical and attractive technology with lot of benefits in field of specializations where human being and digital world are interconnected. The AR applications typically are computing oriented which means lot of data is exchanged and the processor power resulting in battery intensive operations and high-performance infrastructure components [4][5]. The delay or lagging during the running of augmented reality applications can void the purpose of technology usage and spoil the original idea behind these applications. Delays in augmented reality programs lose the point of using technology and provide the end user with a less than ideal experience. AR and VR combine the real world with computer-generated virtual images to create a realistic environment in response to user interaction. As a result of advances in mobile apps, mobile phones, laptops and tablets have become very popular and convenient. The features of these programs and the benefits of their utilization could not be utilized due to complex calculations and severe battery drain [6]. MCC can solve computational and battery charging issues, but lacks high quality streaming and cannot be used in real-time applications. These are time sensitive and require considerable computing power and battery capacity rather than using the full application on mobile devices. To address the issue of latency, a new concept called Mobile Edge Computing (MEC) Energy Consumption Technology has been introduced. In this (MCE) scenario, the user equipment closest to the mobile network edge is shifted. It distributes storage capacity among wireless network access points, enables direct traffic between end users, and provides high-quality AR services. As above discussed about the power consumption of mobile phone clients in computing. The proven technology will enhance on experienced latency for such MEC offerings even as decreasing the overall network usage when getting access to such services. In addition, the replication implemented for video distribution proved six times extra green than preferred IP technology [7]. The successful trial is realized as a natural software program solution over widespread computing hardware and industrial SDN-primarily based switches without requiring a fully-fledged IP community in the vicinity. Latency discount, better bandwidth utilization, and the capacity to deploy such services very near quit customers rather than in a few remote clouds are essential to the success of MEC services, stated Dirk Trussed, Senior Principal Engineer, Interdigital. “This trial showcases the answers, evolved under the leadership of Interdigital that could supply those performance upgrades under realistic conditions and with actual users pleasing the best criteria of trials within the 5G international [8]. Fast battery charging is achieved by increasing the processing power of mobile devices so that they can handle more features without significantly changing the design of the battery [9]. Mobile Edge computing which is now formally known as Multi-Access Edge Computing (MEC) which is considered as the network structure or the cloud extension in which cloud properties are extended and brought to the edge of the computing. The core idea of the MEC is the performance of the processing tasks related to customer cellular jobs and running applications at the customer’s end so that the applications may perform better. The MEC solutions are typically provided at the edge of the networks such as access points and base stations. The solutions are deployed at the base stations and in this way MEC technology is designed to enable and deploy rapidly new applications for the customers. These solutions combine the various properties of telecommunication networks and the information technology. In MEC, the third parties namely content providers and application programmers and developers are given authorities and access to Radio Access Network (RAN) by the mobile cellular operators [10]. Various mobile devices (MD) including mobile phones, vehicle-based computers, game controllers and consoles have been converted to ubiquitous. Many of the applications have grabbed attention such as image processing, face recognition, interactive gaming, natural language processing and augmented reality [11]. These MD device applications are very sensitive and more and more intensive computation is required. This intensive computation needs more energy as such type of the computations are considered as high-energy properties for consumption. As per the constraint size, mobile devices are typically limited in resources i.e the limitations in battery energy and capability in computation [4][5]. As per review of the recent literature, Mobile Edge Computing (MEC or MECO) have given a path to solve such type of problems and effectively cater energy and latency limitations of the mobile devices in which computing related to hardware and computation-intensive jobs are placed to clouds working on the edges of mobile networks for execution [8]. The limited power of computation and energy level make mobile devices impose many of the challenges. To cope these challenges, MEC provides a way to place servers at base stations to harness computing power at the forefront of mobile networks. This technique provides users of mobile devices with a high-quality experience while reducing latency. Industry and scholars are getting a lot of attention to mobile edge computing [12]. Cloud computing is a widely used technology that helps mobile and IoT devices save energy. Cloud computing allows devices to upload computationally complex data to cloud servers. Cloud servers have a great deal of computing power to assist resource constrained devices [13]. However, cloud computing latency is incompatible with many of today's latency-sensitive applications. It can connect other device to a cloud server (LTE) using WiFi or wireless broadband technology such as (LTE). Network energy consumption is projected to decrease due to one factor in 5G architectures [14].

II. PROBLEM STATEMENT

The MEC relief system described in [11] and [12] takes into account the capacity and latency requirements of the MEC server in order to design an energy efficient framework, as shown in Fig 1. However, those designs ignore coordination with nearby free units in the network. In addition, such solutions are not well suited for populated networks because they overload the edge server after a certain number of queries [12]. By including D2D integration in the framework, authors can take advantage of the device features available in the network, reducing the load on the edge server while contributing to further energy savings.

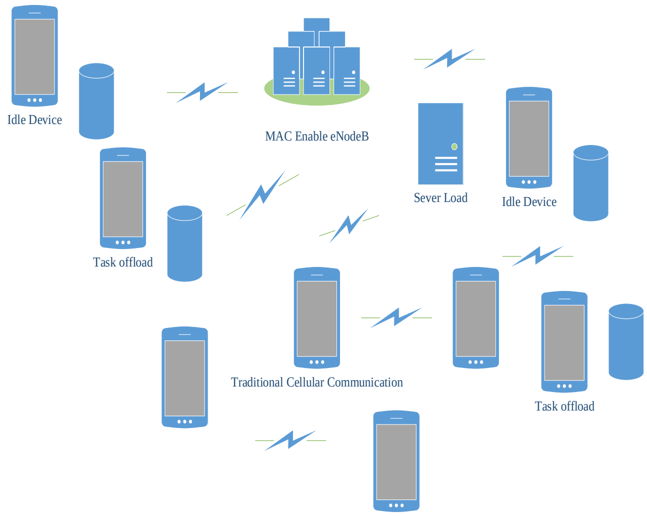

Fig. 1. D2D Offloading Scheme

The main motivation for building a 5G network is to enable low latency applications such as facial recognition. However, current D2D relief techniques [15] and [16] only consider device energy usage and exclude delay-sensitive data. MEC's rescue computing capabilities are also overlooked [17]. Even though it can be used to save power from mobile devices and shorten rescue timeframes. The authors show in Fig 1.1 that the D2D download method offloads the server by completing tasks without using the server's CPU capabilities. In an extended mobile network, node B (eNodeB) is a communication node that connects user devices to the rest of the network. In older networks, it was known as a base station [12]. Because the 5G edge server is located in the eNodeB or base station, you can leverage the resources of the 5G edge server for data mitigation. Also, sending data consumes more power than receiving it. However, they considered it to facilitate investigation and did not differentiate between them. As a result, for data with different Input / Output ratios, those methodologies produce the same result. The authors have found that in terms of energy expenditure and task deadlines, no complete study has been done to compare D2D and MEC as a remedy for task execution.

I. RESEARCH AIM AND OBJECTIVES

The challenge now is to find a way to combine D2D and MEC into a low latency, energy efficient system that overcomes the shortcomings of both options. The main goal of this study is to present a delay-aware, energy-efficient system for carrying out missions using the D2D capabilities of the MEC network. As these two technologies are built into 5G, further research is needed to design systems that allow devices in 5G networks to collaborate effectively. The purpose of the survey is as follows.

(A) Investigate IoT/mobile device energy and latency challenges, as well as current ways for reducing them through D2D collaboration and MEC offload.

(B) Using the D2D collaboration approach in MEC, create an offloading strategy that considers energy and delay characteristics.

(C) In terms of energy efficiency and total run time, compare the proposed approach to present D2D collaboration and MEC offload.

II. RESEARCH METHODOLOGY

This study provided a novel and comprehensive scheme for available device, edge, and cloud server capacity. In this study, before formulating the IoT/user devices, MEC servers, task parameters, and task execution problems, this research first established the network topology and provided a computational rescue selection algorithm. Two methods are presented to find the optimal landing position. One is for time-sensitive data and the other is for devices that have power restrictions. As a result of exhaustive planning, problem characterization, and relaxation of decision-making algorithms, the second objective of the study was achieved. Simulations were used to achieve the ultimate research goal. Many simulation tools were evaluated in this study, including NS-2, NS-3, OMNET ++, MATLAB, and ON. However, no method has been developed to provide the full relief capabilities of MEC in LTE networks, use of batteries, or D2D communications. As a result, Omnit++ simulation tools were created to represent node mobility on genuine LTE channels with D2D communication capabilities.

III. RESULTS

As a result, the study begins with the results of energy savings, the use of mobile devices and the impact of EEDOS on task performance. Finally, this research evaluates various criteria to show that EEDOS is superior to recent studies. If possible, compare EEDOS with the D2D unload schedule and MEC unload schedule. The following measurements were used to analyze EEDOS performance and compare it to previous studies. Examine the performance of EEDOS by verifying the parents of EEDOS by comparing the simulation, development, simulation of D2D collaborative model, simulation of MEC relief settings, and the results of the previous steps. EEDOS saved 97% of energy compared to local data execution. The results show that EEDOS outperforms existing data relief solutions in terms of energy efficiency and data execution time for mobile / IoT devices when compared to edge computing data relief and D2D collaboration. EEDOS requires significantly less server computer capacity than the MEC unloading schedule. Finally, EEDOS reduced the number of missed deadlines compared to previous research.

A. *Energy Saving*

The energy saving rate is shown in simulation iterations, but the number of units is limited to 100 nodes. There is a large logical gap. However, to investigate the impact of work allocation on energy savings, this study increases the chances of performing a task from 30% to 70%. Increasing the task allocation rate in the latter reduced the number of units available in the neighborhood. EEDOS, on the other hand, consistently saves energy in the range of 95.8% to 96.9% in both cases. The D2D mitigation is 50% more energy efficient than when working on local hardware. The average energy efficiency of their system is 60%. The reason for this discrepancy is that according to the timetable, the parameters will be delivered to the edge server when the user completes the allocation. The Edge Server then informs the device that it can complete the task on its own. This study assumed that all devices with a task did not have enough energy to perform the task. As a result, the task cannot be completed and must be delegated to MEC or another adjacent entity.

B. *Energy Consumption*

Fig 2 and 3 show the average energy consumption during an activity. Estimates are based on actual data for smartphones' energy use. Fig 2 shows the increase in the number of units. This study compared four possibilities.

TABLE 1: Simulation Results of Consumption of Energy as the Number of Devices Increases

EEDOS	MEC Offloading	D2D Collaboration	No Offloading
50	110	270	570
50	120	250	570
50	120	220	580
60	140	210	580
60	120	250	580
60	130	300	590
70	140	380	590
80	160	440	590
80	200	490	600

(i) No unloading means that the device completes the task on its own hardware rather than on the entire network. (ii) MEC mitigation: If the device uses less energy, you can perform the task locally instead of downloading it to MEC. (iii) D2D unload. If the MEC determines that unloading saves energy, the device can perform the task locally or unload it to its neighbors. (iv) EEDOS. The device can perform tasks locally, unload to MEC, and offload to neighbors if unloading saves energy.

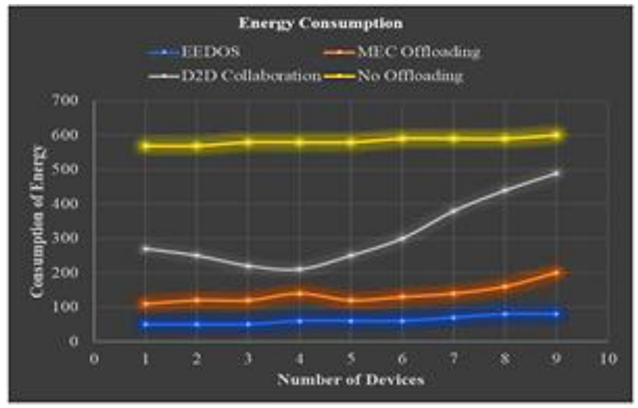

Fig. 2. Consumption of Energy as the Number of Devices Increases

Studies show that EEDOS mobile devices consumed from 21.8mJ to 29.60mJ. Compared to the MEC rescue scheme, which consumes significant energy from 168.5mJ to 349mJ, the D2D rescue scheme worked better from 133.3mJ to 225.3mJ after EEDOS. The use of the computing capabilities of the network's edge servers was the reason for EEDOS's superior performance compared to D2D relief. The performance of the MEC unloading schedule will decrease significantly as the number of units increases. Numerous relief petitions have overloaded the MEC. If MEC is unable to handle additional tasks due to congestion, it is suitable for network-enabled devices to perform and complete the tasks in EEDOS.

TABLE 2: Simulation Results of Increased Task Frequency Leads to Increased Energy Consumption

EEDOS	MEC Offloading	D2D Collaboration	No Offloading
50	190	110	580
50	190	180	570
50	180	190	570
50	180	190	560
40	180	200	560
40	180	200	560
30	170	200	570
30	170	200	570
20	160	190	580

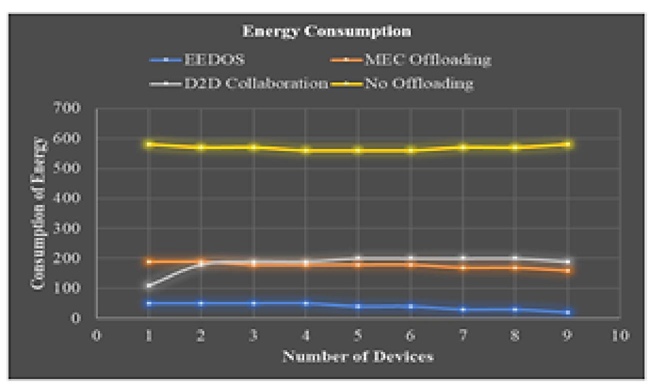

Fig. 3. Increased Task Frequency Leads to Increased Energy Consumption

Fig 3 shows the same energy consumption metric for mobile devices, but with enhanced data distribution for a fixed number of network devices. As shown in both images, completing the process on local hardware without unload instructions consumes significantly more energy than other unload schedules (552.9mJ to 593.2mJ).

C. *Execution Time*

As the number of question units rises, Fig 4 depicts the average time required to complete a task (20-100 units).

Table 3: Simulation Results of Delay in Execution as the Number of Devices Grows

EEDOS	MEC Offloading	D2D Collaboration	No Offloading
0.8	1.1	1	1
0.7	1.2	1	1.1
0.8	1.2	1.2	1.1
0.8	1.3	1.2	1.2
0.7	1.3	1	1.2
0.6	1.1	1	1.2
0.6	1	1	1.3
0.9	1.2	1.2	1.2
0.7	1.3	1.2	1.2

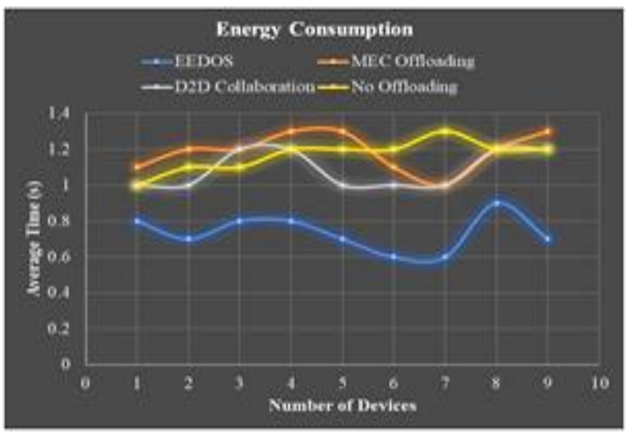

Fig. 4. Delay in Execution as the Number of Devices Grows

In this study, 55% of the units in the network were assigned tasks. As the number of units increased, more free nodes became ready to take part in the collaboration. As a result, when the number of devices exceeds 30, the average execution time of EEDOS will be short. The Simulation of this all the free nodes provide the remaining CPU power to the task. This means that data that is sensitive to MEC integration delays will take exactly 1.0 second, but if the unload destination (free node) has a high processing power, the time required for D2D will be short. increase. In Fig 4, MEC offload performance was the worst because the edge server is holding resources to serve more users.

As a result, the MEC's task took a long time to complete. Because the inactive network node has allocated all CPU resources to the unloaded task after the MEC, D2D unloading has been improved. By adopting available devices rather than MECs, EEDOS outperformed previous systems in terms of execution latency and data transmission time. As explained in the next section, the number of successful EEDOS reliefs has increased compared to the D2D reduction.

Table 4: Simulation Results of Delay in Task Execution When Task Frequency Rises

EEDOS	MEC Offloading	D2D Collaboration	No Offloading
1.2	1.4	1	1.1
1.1	1.4	1	1.1
1	1.3	1.1	1.1
1	1.3	1.1	1
0.9	1.2	1.2	1
0.8	1.2	1.2	1.1
0.8	1.1	1.2	1.2
0.7	1.1	1.3	1.2
0.7	1	1.3	1.1

As a result, by offloading more data, EEDOS reduces overall execution time. Consider the following scenario. The device cannot download the work and it takes a few seconds to perform the task on the local hardware because the D2D schedule does not have the required unload destination. As a result, the overall implementation time is longer than the EEDOS system.

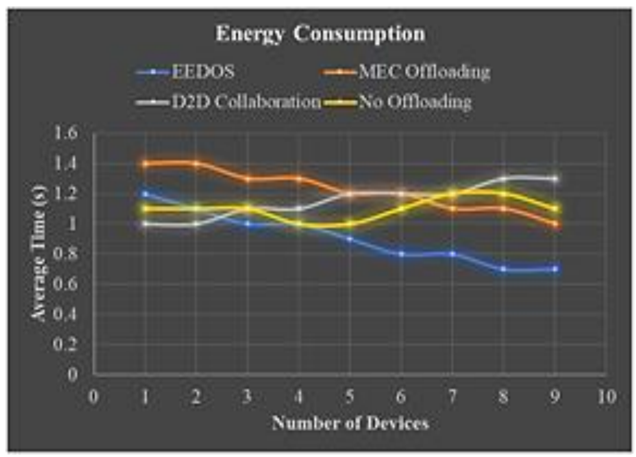

Fig. 5. Delay in Task Execution When Task Frequency Rises

Fig 5 shows the result of execution delay as a function of increasing the task allocation rate for a particular number of 100 nodes in the network. (Range from 20% to 80%). Keep in mind that devices that perform tasks on your local hardware will be used as a baseline in the absence of unload results. If the device does not have enough energy to complete the task, it will not be able to generate output or contribute to the average run time. In addition to reducing energy consumption, it was estimated to demonstrate improved EEDOS execution time.

D. *Successful Offloading*

The number of successful download operations in the EEDOS and MEC download programmes is compared in this section. In this research compared the frequency of task assignments (F = 20, 50 and 80 percent) with the increase in the number of units (10 to 100 units). As the number of devices increases, so does the amount of data supply to EEDOS. Obviously, the number of successful tasks in EDDOS grows with the number of devices. The MEC Relief Plan (shown in red) can only handle up to 18 rescue requests. EEDOS outperforms MEC's remedy because the number of data loads was proportional to the processing power of the edge server when the author simulated the work. As a result, it is no longer possible to accept new requests after 18 issues of resource redistribution for relief. EEDOS supports integrated D2D collaboration for data execution. As a result, free nodes near the required device help complete the operation and increase the chances of a successful unload.

E. Meeting Deadline

Fig 6 shows the number that missed the deadline. This means that the number of runs has exceeded the time limit due to unsuccessful unloading and lack of sufficient processing capacity. As mentioned earlier, the author does not provide an offload value to compare with the offload algorithm. Local execution activity is inferior to EEDOS and D2D solutions that take advantage of the computing power of multiple network devices (assuming all devices have enough energy to complete the task). According to studies, EEDOS is the best at fulfilling deadlines. D2D relief missed three and MEC relief missed four. MEC relief misses 12 deadlines for every 100 nodes in this network, whereas D2D relief increases performance with only 7 errors. EEDOS gave the best result that 50% of network nodes assigned tasks after only 5 missed deadlines. D2D relief is superior to MEC relief because it leverages the computing power of many open network devices, but MEC relief utilizes only the computing capabilities of the edge server. As the number of devices on this network increases more likely to find a nearby free device that can complete the unloaded operation.

Table 5: Simulation Results of Missed Deadlines

EEDOS	MEC Offloading	D2D Collaboration	No Offloading
0	4	3	3.5
1.5	5.5	4	4.5
1.5	6	4.5	5
2	7	5	6
2	7.5	5.5	7.5
4	9	6	8
4	10	6	8.5
5	11	7.5	9
5.5	12	7.5	10

As a result, the EEDOS and D2D remedies are clearly superior to the MEC remedies. Free device functionality is built into EEDOS's network and edge servers, increasing the network resources that can be useful in rescue operations. As a result, when compared to other methodologies, EEDOS gave the best results in terms of allocation completion.

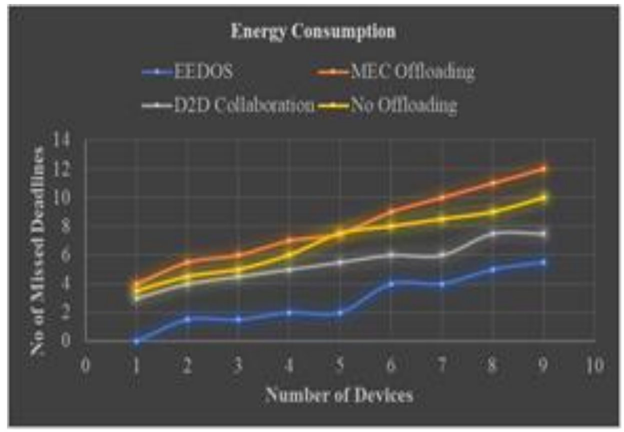

Fig. 6. Missed Deadlines

IV. CONCLUSION

This research does not look at communication about physical inventory. It is important to note that the aim of this research was not to improve D2D or MEC's energy efficiency. Drawback was that the EEDOS system static and did not take into account the mobility of mobile devices. In addition, the authors expected that a nearby dormant unit would be happy to help with unloading. Despite the fact that this is a common concept in research, it is not a viable option. Users are more interested in their energy in the real world and need to implement incentive systems to facilitate collaboration.

REFERENCES

[1] Saleem, U., Liu, Y., Jangsher, S., Tao, X., & Li, Y. (2020). Latency minimization for D2D-enabled partial computation offloading in mobile edge computing. IEEE Transactions on Vehicular Technology, 69(4), 4472-4486.

[2] Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T. (2018, feb). Mobile Edge Computing: A Survey. IEEE Internet of Things Journal, 5(1), 450–465

[3] Ahmad, H., Agiwal, M., Saxena, N., Roy, A. (2018, aug). D2D-based Survival on Sharing for critical communications. Wireless Networks, 24(6), 2283–2295.

[4] Kai, C., Zhou, H., Yi, Y., & Huang, W. (2020). Collaborative cloud-edge-end task offloading in mobile-edge computing networks with limited communication capability. IEEE Transactions on Cognitive Communications and Networking, 7(2), 624-634.

[5] Fan, W., Han, J., Yao, L., Wu, F., & Liu, Y. A. (2020). Latency-energy optimization for joint WiFi and cellular offloading in mobile edge computing networks. Computer Networks, 181, 107570.

[6] Ranji, R., Mansoor, A. M., & Sani, A. A. (2020). EEDOS: an energy-efficient and delay-aware offloading scheme based on device-to-device collaboration in mobile edge computing. Telecommunication Systems, 73(2), 171-182.

[7] Hossain, M. S., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2020). Edge computational task offloading scheme using reinforcement learning for IIoT scenario. ICT Express, 6(4), 291-299.

[8] Wang, Q., Guo, S., Liu, J., & Yang, Y. (2019). Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. Sustainable Computing: Informatics and Systems, 21, 154-164.

[9] Xiao, S., Liu, C., Li, K., & Li, K. (2020). System delay optimization for mobile edge computing. Future Generation Computer Systems, 109, 17-28.

[10] Guo, M., Li, Q., Peng, Z., Liu, X., & Cui, D. (2022). Energy harvesting computation offloading game towards minimizing delay for mobile edge computing. Computer Networks, 204, 108678.

[11] Chen, X., Cai, Y., Li, L., Zhao, M., Champagne, B., & Hanzo, L. (2019). Energy-efficient resource allocation for latency-sensitive mobile edge computing. IEEE Transactions on Vehicular Technology, 69(2), 2246-2262.

[12] Li, Z., Zhu, N., Wu, D., Wang, H., & Wang, R. (2021). Energy-efficient mobile edge computing under delay constraints. IEEE Transactions on Green Communications and Networking, 6(2), 776-786.

[13] Zhao, M., Yu, J. J., Li, W. T., Liu, D., Yao, S., Feng, W., ... & Quek, T. Q. (2021). Energy-aware task offloading and resource allocation for time-sensitive services in mobile edge computing systems. IEEE Transactions on Vehicular Technology, 70(10), 10925-10940.

[13] Park, J., Solanki, S., Baek, S., & Lee, I. (2021). Latency minimization for wireless powered mobile edge computing networks with nonlinear rectifiers. IEEE Transactions on Vehicular Technology, 70(8), 8320-8324.

[14] Yang, T., Feng, H., Gao, S., Jiang, Z., Qin, M., Cheng, N., & Bai, L. (2019). Two-stage offloading optimization for energy–latency tradeoff with mobile edge computing in maritime Internet of Things. IEEE Internet of Things Journal, 7(7), 5954-5963.

[15] Dab, B., Aitsaadi, N., & Langar, R. (2019, April). Joint optimization of offloading and resource allocation scheme for mobile edge computing. In 2019 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-7). IEEE.

[16] Feng, W., Liu, H., Yao, Y., Cao, D., & Zhao, M. (2021). Latency-aware offloading for mobile edge computing networks. IEEE Communications Letters, 25(8), 2673-2677.

[17] Feng, J., Pei, Q., Yu, F. R., Chu, X., & Shang, B. (2019). Computation offloading and resource allocation for wireless powered mobile edge computing with latency constraint. IEEE Wireless Communications Letters, 8(5), 1320-1323.