# Efficient Task Scheduling and Offloading for UAVNetworks using Nash and ML Approach

[1]Dilip S, [2]Dharamendra Chouhan, [3]Anand R Umarji, [4]Chaitra B V,[5]Sanjay

Department of Computer Science and Engineering

University of Visvesvaraya College of Engineering (UVCE - on IIT Model), Bangalore, India

{dilipgs9916@gmail.com, dharmu2007@gmail.com, ru.anand@gmail.com, sanjaygodbole7543@gmail.com}

**Abstract**

*With the increasing number of Internet of Things (IoT) devices, effective computing performance has become a critical issue. Strong processing power is required for a multitude of Internet of Things applications, including traffic control, augmented reality, location tracking, and autonomous driving, which all need extensive real-time data processing. The introduction of Mobile Edge Computing (MEC), aims to safely and effectively tackle this issue via the internet. IoT devices may now be used to offload computationally demanding activities by including a MEC server. Delays and transmission cost, however, are significant disadvantages. By serving as MEC servers, Unmanned Aerial Vehicles (UAV), could potentially lessen this problem thanks to their great mobility and inexpensive cost. No matter where a user is, mobile networks offer wireless access. These networks may transmit and receive data utilising neighbour connectivity, and they are self-configurable. Sensors, mobile devices, routers, and many other tiny devices can be used to create this network. However, because these devices are small and unable to perform complex computations, the author of this paper uses 5G enabled UAV based community offloading, inwhich UAVs move to various positions and mobile devices offload or schedule tasks to the closest freestanding (UAV with less load) UAV. After receiving a job, the UAV will schedule it to be sent to a task processors or base station, which will process it and return the results to the UAV, which will then transmit it toa mobile device.*

**Keyword**-- *NASH algorithm, multitasking, mobile edgecomputing, and unmanned aerial vehicles.*

## I. Introduction

The world has drastically changed as a result of the proliferation of smart gadgets, and cellular technologies are now widely used. Social networking sites and online streaming services have grown in popularity and use for people of all ages. Cisco projected in 2020 that by the end of 2021, cellular data traffic will have increased seven times(Cisco & Internet, 2020). The rise in data that followed hasput a heavy pressure on mobile service providers. Cellular networks will becoming even more crowded without adequatesafeguards for maintaining and processing such their workloads, which would lead to worse quality and lower download speeds. For this reason, mobile devices require more processing power. Additionally, as 5G technology develops over time, a variety of services that need complex processing tasks—tasks that the specified devices are not capable of handling—have been inspired.

The idea of the metaverse, which has been called the mobile Internet's their [1], is becoming more and more well- liked. It is the Internet materialised, with a human-centered, immersive, and interconnected virtual environment that users may travel with virtual characters under their control [2]. Users may engage with the virtual environment in the metaverse by moving their bodies or making sounds, and they can take use of a wide range of human-centered virtual services for instance, people can employ technologies likevirtual reality, augmented reality, and mixed reality to explore another dimension in the metaverse. Nonetheless, the salient feature of these innovations is their ability to produce pictures in a timely manner, so producing perceptual visuals in response to users' ideas. This necessitates devices to possess a high processing capacity and ample energy. The increasing advancement of technology connected to the metaverse and users' need for dynamic, real-time virtual world perception prevent standard wired devices from providing ubiquitous network connectivity across the metaverse. Instead, users mayconstruct a dynamic, human-centric virtual environment that is aware of their requirements in real time by running metaverse programmes on their mobile devices [1]. For amore lifelike experience in the metaverse, each user createshis or her own image and communicates with other avatars and virtual items at any time and place. This naturally places aheavy burden on each terminal's processing power and battery life.

Currently, mobile devices' computational power and batterylife are constantly limited due to cost of production and

technological constraints [3]. Recent years have seen the rise of Mobile Edge Computing, also known as MEC, [4] as a viable option for real-time augmented and virtual video delivery over wireless networks and as a critical component for real-time rendering.

It efficiently increases the processing capacity of mobile devices by transferring calculation tasks to nearby MEC servers [5]. Additionally, as Wireless Voltage Transfer (WPT)technology has advanced, mobile devices' energy limitations have decreased. By combining WPT and MEC, Wireless Power Mobile Electronics Computing (WPMEC) enhances the computing capability to serve the metaverse whilereducing the impact on battery life [6]. The term "wireless powered metaverse paradigm" describes how WPMEC technology makes it simple to run human-centric programmes designed for the metaverse on mobile devices.

Through process offloading, mobile edge computing, orMEC, is a potential approach that makes use of cloud servers set up to assist mobile devices. The original idea for edge computing, or cloudlet, was put out in 2009. Although cloudlets let mobile users access cloud services, they force users to switch between cellular and Wi-Fi networks.

By leveraging the concept of an on-demand cloud, handheld devices can also be used to do activities locally. By combiningthe computing capacity of several user devices, it enables the processing of tasks. Specifically, shifting the information to aninterface computer improves user experience and prolongsbattery life. Cisco first proposed the concept of computation offloading in 2012. Consequently, processing tasks may be wirelessly transferred from any low-resource mobile device tohigher-resource devices. The other machines transmit theirresults back to the portable devices after they have completed their tasks. Unfortunately, this technique still falls short of expectations because of the limitations of Bluetooth technology in mountainous and rural areas. Additionally, emergency response instances should always come first. Therefore, it is challenging to maintain energy economy and experience quality without sacrificing communication speed, even in the best of circumstances.

This essay's remaining sections are organised as follows: InSection II, we offer pertinent literature; in Section III, we construct the system model and delineate the optimisation problem. In Section IV, the two-phase alternating optimisationapproach is devised, and in Section V, its effectiveness isevaluated. Section VI finally concludes this endeavour.

## II. LITERATURE SURVEY

The concept of computation offloading is quite popular,and several surveys have highlighted different facets of offloading in edge computing settings. The authors ofReference. [1] addressed how UAVs may be used as edge servers in offload and gave an overview of current developments, unresolved problems, and potential application possibilities of UAVs in MEC contexts. The writers of Ref.[2]provide a summary of a cutting-edge computational job offloading method by emphasising performance measures, including energy consumption reduction, to guarantee quality-of-service while implementing resource allocation protocols. In Ref., offloading according to certain application objectives was examined [4]. Based on previously published research,the authors there gave a thorough overview of the traffic and compute offloading duties. Ref examined methods for machine-learning computation offloading (Shakarami et al., 2020a). whereby the writers evaluated the methods, features, performance measures, strengths, and shortcomings of the employed methodologies against the literature.

In contrast to such studies, offloading has been studied to overcome issues in dynamic situations (e.g., automotive). Using a classification of communication channels and design aims, recent experiments described in the published literature were summarised by Ref[6]. Ref's authors [5] examined fundamental models from the viewpoints of energy harvesting, computing, and communication. In Ref., an overview of recent advancements in machine-learning approaches was given [9]. The authors of Reference. [7] examined how an offloading system may adjust to enable the building of aneven more reliable and scalable capability while maintaining the level of experience. The writers of Ref [8] paid closeattention to the stochastic behaviours of offloading strategies made possible by variations in mobile applications. The authors talked about edge computing, fog, mobile cloud, and other related settings as well as stochastic offloading strategies.The paper's authors of Ref. [12] conducted a thorough assessment in which they examined current advancements in edge computing from the viewpoints of computation andarchitectural Offloading choices, which are based on allocation of resources and mobility management, werefurther examined. Ref. [13] looked at an unloading mechanism based on game theory and a special ontology. There, key measurements, application situations, applicable techniques, and tools related to loading processes in a MEC environment were discussed by the authors. In the setting of channel access strategies, the authors of [14] provide a brief synopsis of the UAV-enabled MEC system and outline energy-efficient techniques for controlling resources andcomputation offloading on IoT devices. UAV-assisted transmission was not considered in this study. The authors of

[15] offered a complete analysis of machine learning and deepanalytics based methodologies in UAV-enabled MEC networking. This review, however, emphasises on artificial intelligence-based methods and ignores the commonly utilisedoptimisation techniques in UAV-MEC design.

The aforementioned studies highlight the methods ofcomputation offloading in a context of edge computing from

different angles. One of the main disadvantages of the typical MEC server is that its location typically remains the same [18] and it is unable to be moved to follow mobile users. In situations like these, unmanned aerial vehicles (UAVs) present a viable option since they may serve as a possible MEC server, offering resources for computingand communication in the event that a ground MEC network iseither non-existent or has been devastated by a natural disaster.Owing to its mobility, quick execution, and low cost, UAVs may be easily utilised in emergency rescue missions, army surveillance, and arid locations. For this reason, a thoroughanalysis of the compute offloading strategies used in UAV- MEC networks is both very intriguing and crucial.

## III.SYSTEM MODEL

This part presents the developed system model and formulatesthe challenge of maximising long-term computing efficiency. Because no unloading facility was available for any of theexisting approaches, their throughput would decrease and their energy consumption would increase. Thus, the author of the proposal paper suggests using UAV connection to offload andschedule tasks to a powerful processing unit like a task processor.

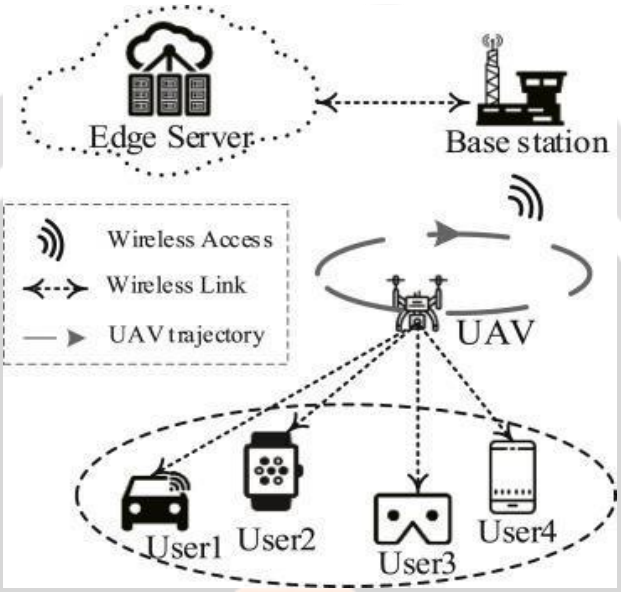



**Fig 1: System Model**

The following modules are what we have combined to unload here.

1) Trajectory: In the paper on trajectory, the machine learning method QLEARNING is discussed. This technique locates UAVs and mobile nodes, whichmight be useful in identifying nearby mobiles and grouping them into a single cluster.
2) 5G Supported UAV Connectivity: In this study, communities are formed through the use of clusters. Mobile devices that are closer to one another aregrouped together to form a single cluster, and then UAVs are allocated to each cluster. The UAV will beassigned to the cluster closest to it.
3) Nash equilibrium task scheduling: To offload a job from any mobile device, we first generate a list of all UAVs that are accessible. Next, we use the NASH algorithm to pick the UAV that is closest to the mobile device, has the most available energy, and hasthe least delay. Thus, by using this technique, free UAVs will always be chosen, resulting in low latency, high throughput, and low energyconsumption.

Since we lack smartphones or unmanned aerial vehicles (UAVs), we conducted the project virtually, and the results of that simulation are the modules listed below.

1) Create UAV Community: This module will be usedto establish a network community.
2) Determine Cluster Size: Using the size of thenetwork, we will determine how many clusters are appropriate to split into groups, with one UAV being assigned to each group.
3) Create Clusters: A group will be formed by the network depending on the number of clusters, and nearby mobile devices will be included in the same group or cluster.
4) UAV Selection: We will assign UAVs to each unit orcluster using this module.

5) Offload Task to Computer using UAV: This module source will allow mobile to offload tasks. A freeUAV will then be chosen using the NASH algorithm to offload tasks to the task processor and provide responses to the mobile device.
6) Using this module, we will create an energy usage graph that compares suggested TDTS offloading methods with non-offloading methods.
7) Throughput Graph: This module will be used to display a throughput graph comparing suggested TDTS offloading methods with non-offloading methods.

## IV. ALGORITHM

In this part, we provide a multitask-based, two-stage alternating optimisation technique that can successfully address the two aforementioned subproblems.

A. QLEARNING

An Off-Policy approach for Sequential Difference learning is termed Q-Learning. With sufficient instruction under any -softpolicy, it can be shown that the approach converges to a near approximation of the action-value contour for any target policy with probability less than 1. Q-Learning finds the optimal path of action even when phases are selected using a more scientific or even random strategy. The procedural version of the algorithm is:

```
Initialize Q(s, a) arbitrarily
Repeat (for each episode):
   Initialize s
   Repeat (for each step of episode):
      Choose a from s using policy derived from Q
         (e.g., ε-greedy)
      Take action a, observe r, s'
      Q(s, a) <-- Q(s, a) + α [r + γ max_α Q(s', a') - Q(s, a)]
      s <-- s';
   until s is terminal
```

The following parameters are utilised for updating the Q-value:

$\alpha$ - The learning rate, which is set between 0 and 1, is one of the factors utilised in the Q-value updating process. Nothing islearnt if it is set to 0, as the Q-values cannot be changed. A high score, like 0.9, indicates that learning may happen rapidly.

$\gamma$ - discount factor, with a range of 0 to 1. This simulates theidea that rewards in the future have less value than those in thepresent. In mathematical terms, the method can only convergeif the discount factor is smaller than 0.

$\max_\alpha$ - the highest reward that may be obtained in the state that comes after the present one; that is, the reward for choosing the best course of action in that state.

The following are the stages that this procedural technique may be converted into in plain English:

1. Create the Q-values surface, Q(s, a), from scratch.
2. Note the situation as it is, s.
3. Using one of the actions selection rules (-soft, -greedy, or softmax) described above on the previous page, pick an actionto perform a, for that state.
4. Execute the action and note the new state, s', and the reward,r.
5. Using the measured reward and the highest reward that maybe earned in the future state, adjust the value of Q for the current state. The forumla and the previously mentioned settings are followed while upgrading.
6. Once the terminal's state is achieved, set the state into the new state and continue.

B. Nash equilibrium

Each participant in a noncooperative action can optimise their outcome depending on the decisions made by the other participants, according to the idea of Nash equilibrium in game theory. The game is considered to have achieved its conclusion when no one feels

motivated to alter their own strategy, even if they're aware of the intentions of the other players. Nash equilibrium.

1) The Equivalent Matrix Game and Nash Equilibrium

Let Xi represent player I's pure strategy set in game Γ. X is used to denote ∏i ∈IXi. In the game of resource allocation,

$$X_i = m_i: = {}^{K_i + N - 1}_{N - 1}.$$

Assume that si, or the mixed strategy, is a likelihooddistribution on Xi for player i.

$$s_i x_i \geq 0, \quad \sum_{x_i \in X_i} s_i x_i = 1.$$

Let S := ∏i ∈ISi be the set of randomised- strategy profiles, and allow Si represent the set of combined approaches for player i. The anticipated value of player I's payout function fors ∈S is

$$\pi_i s = \sum_{x \in X} \pi_i x \prod_{j \in I} s_j x_j.$$

Nevertheless, since interactions between players are pairwise,we may express πi(s) in a more straightforward manner as follows:

$$\pi_i s = \sum_{j \in I} \pi_{ij} S_i S_j,$$

Where

$$\pi_{ij} s_i s_j: = \sum_{x_i \in X_i, x_j \in X_j} \pi_{ij} x_i x_j s_i x_i s_j x_j.$$

For each ti ∈Si, we let (s-i, ti) signify the randomized- strategy profiles for which the i-th element is ti and all otherparts are as in s. Consequently,

$$\pi S_{-i} t_i = \sum_{x \in X} \prod_{j \in I \setminus i} S_j x_j \, t_i x_i \pi_i x,$$

or

$$\pi S_{-i} t_i = \sum_{j \in I} \sum_{x_i \in X_i, x_j \in X_j} \pi_{ij} x_i x_j t_i x_i s_j x_j.$$

### C. Shortest Distance First:

SDF typically refers to a strategy or algorithm that prioritizes tasks, movements, or routes based on minimizing the physical distance. This concept is commonly applied in various fields, such as logistics, robotics, and networking. In logistics and transportation, SDF might involve prioritizing delivery routes or scheduling transport vehicles to minimize the total travel distance. This helps in optimizing fuel efficiency, reducing travel time, and improving overallresource utilization.

Particularly for mobile robots or drones, the SDF principle guides the planning of motion trajectories. The goal is to minimize the distance travelled by the robot while navigating through an environment, which is crucial for efficient and timely completion of tasks.

In the context of task scheduling, SDF might be applied to prioritize the execution of tasks based on their physical proximity. This can be relevant in scenarios where completingtasks in close proximity to each other is more efficient, suchas in manufacturing processes.

In general, the "Shortest Distance First" approach is often associated with optimization strategies that aim to minimize the overall distance traveled or path taken, leading to resource-efficient and timely solutions.

### D. Trajectory Design and Task Scheduling Algorithm(TDTS):

The existing non-offloading SDF technique, emphasizing the challenges or limitations related to throughput. TDTS to enhance throughput in a non-offloading SDF) context.

TDTS schemes involve dividing time into discrete slots andallocating these slots to different communication channels or users. Each channel or user gets exclusive access to the communication medium during their assigned time slot. This type of approach is often used to manage the sharing of a communication medium efficiently. The total time frame is cyclic, and the cycle repeats.

During their assigned time slot, a channel or user has exclusive access to the communication medium. They cantransmit their data during this time without interference from other channels or users. Since each channel or user has a dedicated time slot, the chance of collisions (simultaneous transmissions) is reduced. Collision avoidance mechanisms may be implemented to handle potential conflicts, such as retransmission protocols. TDTS aims to improve the efficiency of communication by reducing collisions and allowing each channel/user to transmit without contention during their allocated time slot.

Throughput is enhanced as multiple channels or users can share the same communication medium without interfering with each other.

### E. Cluster Algorithm

When it comes to UAV offloading tasks to a community of ground-based devices, clustering algorithms play a crucial rolein organizing and optimizing the communication and coordination between UAVs and the devices in the community. Clustering helps manage the workload efficiently and distribute tasks among UAVs and ground devices.

**Working**

- Devices in the community register themselves to the UAV network, providing information about their capabilities, available resources, and current workload.
- UAVs are initialized with their communication range, available resources, and the types of tasks they can offload. Each UAV receives information about the tasks that need offloading. This information includes task types, computational requirements, and deadlines.
- Group devices into clusters based on their geographical proximity to UAVs. This helps in minimizingcommunication latency and optimizing energy consumption for both UAVs and devices.
- Distribute the tasks evenly among UAVs within a clusterto balance the computational load. This ensures that no UAV is overloaded while others have spare capacity.
- Establish and maintain real-time communication linksbetween UAVs and devices within each cluster. This is essential for task coordination, status updates, and dynamic adjustments based on changing conditions.
- Periodically reassess the workload and resources in each cluster. If there are changes in task requirements or the availability of devices, dynamically reconfigure the clusters to adapt to the evolving environment.
- Assign priorities to tasks based on their urgency, deadlines, and importance. Ensure that high-priority tasks receive appropriate resources and are offloaded promptly.

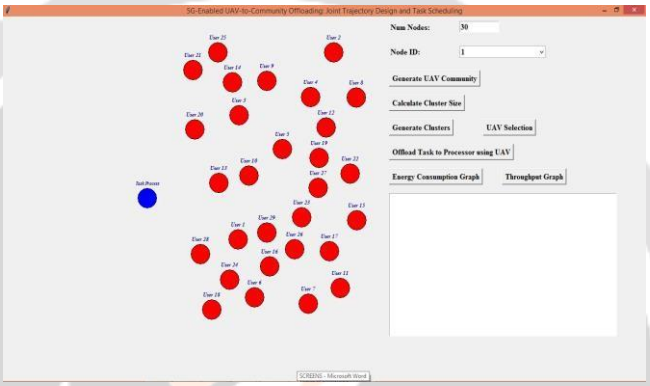
- Implement fault tolerance mechanisms to handle UAV ordevice failures. Reassign tasks to available resources andadjust the clusters accordingly.
- Consider energy constraints for both UAVs and devices. Optimize the clustering and task assignment to minimize energy consumption, especially for UAVs with limited battery capacity.

The effectiveness of the clustering algorithm depends onthe specific requirements of the UAV-to-Community offloading scenario, including task characteristics, communication constraints, and the capabilities of UAVs and ground devices. The algorithm may need customization based on the unique aspects of the application domain.

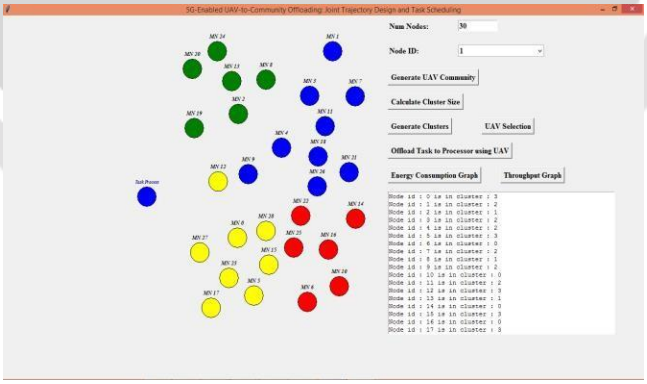
## V. EXPERIMENT RESULTS

Enter the number of nodes in the Num Nodes field below, and then click the "Generate UAV community" button to get the output shown below.

After entering "Num Nodes" as 30, the simulation screen displayed red and blue circles, representing community members in the red circles and task processors in the blue circles. Click "Calculate Cluster Size" to organise all users into that number of clusters.



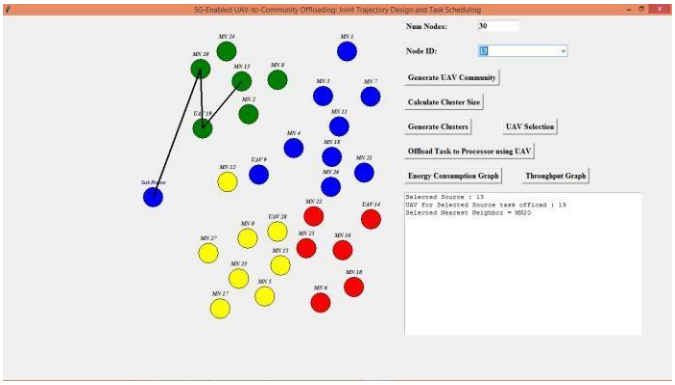

The same colour nodes in the screen below are in onecluster, and the other colour nodes are in a different cluster or community. Click on "UAV selection" to designate a node asa UAV with high energy and coverage of a greater number of nodes in the community. Each community will receive one UAV, and the output shown below will come from it.



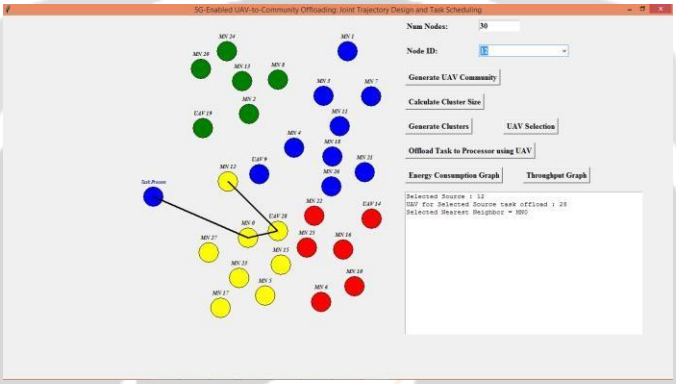

To offload the work and obtain the results below, choose source number 7 from the list then click the "Offload Task to Processors using UAV" button.

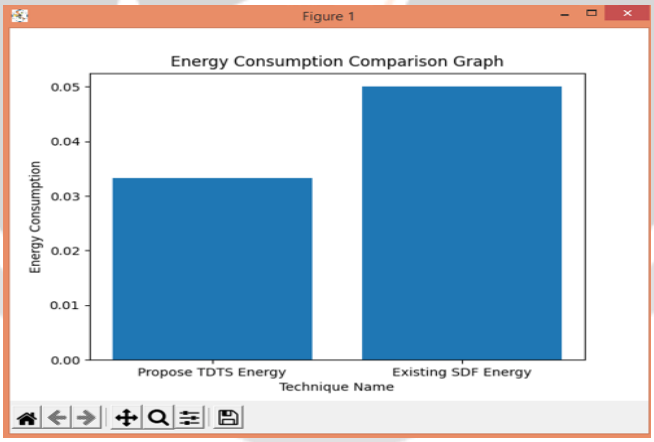
Similar to the below screen, you may choose another source to offload tasks. In the screen below, user 7 is offloading a job to its group UAV 9, which is dumping to a task processor.

The work that MN13 is transferring to UAV 19 is then offloaded to the task processor.

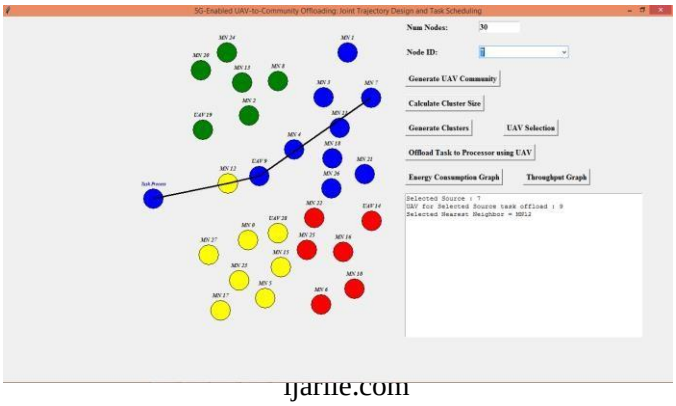


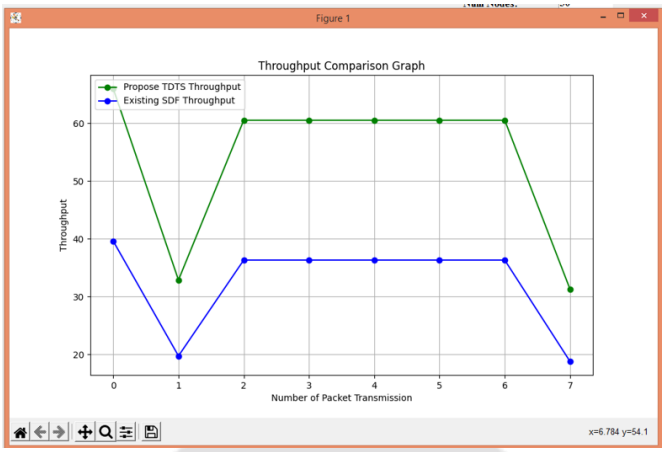

Tasks are being offloaded by MN12 to its community unmanned aerial vehicle (UAV28), which is then offloaded to a task processor. Click the "Energy Consumption" graph to view the graph below.




The y-axis in the energy consumption graph above shows the names of the algorithms, and the x-axis shows the amount of energy consumed. The proposed TDTS approach uses less energy than the current non-offloading technique. Click the "Throughput Graph" button to view the graph below.






The following throughput graph shows the number of packets sent on the x-axis, the throughput on the y-axis, the proposed TDTS throughput shown by the green line, and the current Non-offloading SDF approach represented by the blue line. The proposed TDTS technique achieved a high throughput in both cases.

## VI. CONCLUSIONS

In this work, we carried out a computation offload in a MEC environment supported by UAVs. We categorised the existing offloading algorithms based on their two methods, Nash Equilibrium and Q-learning. Next, we conducted a comparison analysis of them based on their unique characteristics, advantages, and disadvantages. An UAV- enabled MEC server would be very helpful and easy to enhance their computing powers in locations where building an infrastructure is fairly difficult, and it should offer smooth connectivity in combat and disaster-prone areas. This study's comparisons allow for a better selection of offloading systems.We also discussed important takeaways and proposals for further study. Energy efficiency and UAV power are critical challenges.. Therefore, it is necessary to build an offloading strategy that takes user mobility and changeable network circumstances into account. We think that the results of this study will be useful in the future for designing and implementing effective offloading strategies for UAV–MEC systems.

## REFERENCES

[1] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. Shen, and C. Miao, "A full dive into realizing the edgeenabled metaverse: Visions, enabling technologies, and challenges," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 656– 700, Nov. 2023.

[2] R. Hare and Y. Tang, "Hierarchical deep reinforcement learning with experience sharing for metaverse in education," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 4, pp. 2047–2055, Apr. 2023.

[3] Y. Zhan, S. Guo, P. Li, and J. Zhang, "A deep reinforcement learning based offloading game in edge computing," IEEE Transactions on Computers, vol. 69, no. 6, pp. 883–893, Jun.2020.

[4] S. Pan, P. Li, C. Yi, D. Zeng, Y.-C. Liang, and G. Hu, "Edge intelligence empowered urban traffic monitoring: A network tomography perspective, IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 4, pp. 2198–2211, Apr. 2021.

[5] Q. Xu, Z. Su, K. Zhang, and P. Li, "Intelligent cache pollution attacks detection for edge computing enabled mobile social networks," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 3, pp. 241–252, Jun. 2020.

[6] X. Wang, J. Li, Z. Ning, Q. Song, L. Guo, S. Guo, and M. S. Obaidat, "Wireless powered mobile edge computing networks: A survey," ACM Comput. Surv., vol. 55, no. 13s, Jul. 2023.

[7] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5723–5728, May. 2020.

[8] Z. Ning, Y. Yang, X. Wang, Q. Song, L. Guo, and A. Jamalipour, "Multiagent deep reinforcement learning based UAV trajectory optimization for differentiated services," IEEE Transactions on Mobile Computing, pp. 1– 17, Sep. 2023, doi: 10.1109/TMC.2023.3312276.

[9] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Stable online offloading and trajectory control for UAV-enabled MEC with EH devices," in Proc. IEEE GLOBECOM, Dec. 2021, pp. 01–07.

[10] W. Feng, J. Tang, N. Zhao, X. Zhang, X. Wang, K.-K. Wong, and J. A. Chambers, "Hybrid beamforming design and

resource allocation for UAV- aided wireless-powered mobile edge computing networks with NOMA," IEEE Journal on Selected Areas in Communications, vol. 39, no. 11, pp. 3271–3286, Nov. 2021.

[11] W. Liu, S. Zhang, and N. Ansari, "Joint laser charging and DBS placement for drone-assisted edge computing," IEEE Transactions on Vehicular Technology, vol. 71, no. 1, pp. 780–789, Jan. 2022.

[12] X. Hu, K.-K. Wong, and Y. Zhang, "Wireless-powered edge computing with cooperative UAV: Task, time scheduling and trajectory design," IEEE Transactions on Wireless Communications, vol. 19, no. 12, pp. 8083–8098, Dec. 2020.

[13] Z. Xu, K. Wu, Z. Che, J. Tang, and J. Ye, "Knowledge transfer in multitask deep reinforcement learning for continuous control," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 15 146– 15 155.

[14] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust multitask reinforcement learning," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4499–4509.

[15] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with PopArt," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 3796–3803.

[16] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 4, pp. 1198– 1207, Apr. 2018.

[17] Z. Zhang, F. Zhuang, H. Zhu, C. Li, H. Xiong, Q. He, and Y. Xu, "Towards robust knowledge graph embedding via multi-task reinforcement learning," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 4, pp. 4321–4334, Apr. 2023.

[18] Q. Qi, L. Zhang, J. Wang, H. Sun, Z. Zhuang, J. Liao, and F. R. Yu, "Scalable parallel task scheduling for autonomous driving using multitask deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 69, no. 11, pp. 13 861–13 874, Nov. 2020.

[19] T. Dong, Z. Zhuang, Q. Qi, J. Wang, H. Sun, F. R. Yu, T. Sun, C. Zhou, and J. Liao, "Intelligent joint network slicing and routing via GCNpowered multi-task deep reinforcement learning," IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 2, pp. 1269– 1286, Jun. 2022.

[20] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "IRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks," IEEE Internet of Things Journal, vol. 6, no. 4, pp. 7011–7024, Apr. 2019.