# Resource-Optimized Multi-Armed Bandit Based Offload Path Selection in Edge UAV Swarms

Anandarup Mukherjee, *Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE,* Vadde Santosha Pradeep Chandra, and Mohammad S. Obaidat (Fellow, IEEE)

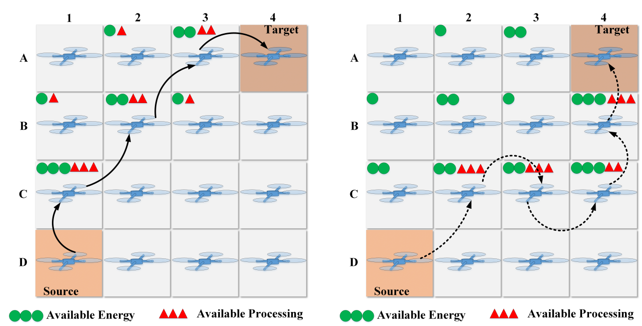
**Abstract**—This work looks into the problem of a decentralized data offloading within an edge UAV swarm to mitigate the complexities of a single UAV continually generating and processing large application-specific data. The mobile edge UAVs considered here are multi-rotor types having constrained energy and processing power, which makes long-term handling of large data volumes impossible for standalone UAVs. The load mitigation is carried out by offloading data from a source UAV to other swarm members with sufficient energy and processing requirements. In this work, we focus on selecting the most optimal multi-hop path through the UAVs concerning available energies and processing resources, which can survive the duration of the data offload between the source and a target UAV. We formulate a Multi-Armed Bandit (MAB) based offload path selection scheme, which selects the most energy and processing optimized multi-hop path between a source and a target UAV. Upon comparison of our scheme against the naive shortest path approach, we observe that our approach results in significant savings of collective network energies, even for long operational durations.

**Index Terms**—Edge UAV network, UAV swarm, Data offloading, Multi-armed Bandits, Reinforcement Learning, Path Selection

## 1 Introduction

Unmanned Aerial Vehicles (UAVs) have found widespread use in aerial search and tracking of ground-based targets in domains such as surveillance, disaster management, border security, wildlife monitoring, and remote mapping [1]. Various approaches have been devised for standalone as well as collaborative UAV swarm based tracking of the ground as well as aerial targets. These tasks produce a massive amount of spatiotemporal data which is processed to get additional information. In contrast, the resource-constrained nature of these aerial platforms has led to the rise of collaborative and autonomous UAV swarms for accomplishing tasks more efficiently and speedily. The decentralized nature of these swarms regarding processing requires advanced computational schemes for processing the generated data within the edge (swarm) without the need of forwarding this data to higher layers along the IoT architecture (Fog and Cloud).

In our scenario, we consider an ad-hoc edge UAV swarm made up of hovering multirotor UAVs. The edge swarm of UAVs is capable of intra-member communication in a multi-hop manner enabling them to perform tasks collaboratively and act as a single unit. The motion of the swarm members is restricted to their grids, and each UAV is considered quasi-static within the grid as represented in Fig. 1. The UAVs are kept in a hovering state to accommodate for tasks such as on-spot video-based situational awareness. The data generated by some of the tasks such as aerial photography and video surveillance are massive in size, which increases with the quality of data. However, despite developments in low-power processing and sensing solutions, UAVs, especially multi-rotor types, rapidly deplete their energy sources to keep themselves airborne. These energy constraints dictate the flight time of these edge UAVs.




(a) Data offload at $t_1$ (b) Data offload at $t_n$

Figure 1: Placement of edge UAVs in the grid-map and selection of suitable data offload path between source and target UAVs.

In order to achieve much faster processing and conserve network bandwidths, intra-edge swarm processing is considered, where data captured from a source UAV might be transmitted to a requesting target UAV by using the multi-hop swarm communication. The collaborative task completion ability of the swarm members proves beneficial for accomplishing assigned tasks collaboratively in an energy efficient and processor-friendly manner, without the need for over-clocking any of the swarm members. In this work, we formulate an edge UAV utility function, which takes into account the energy, processing, and communication distance requirements of these UAVs. In continuation, we propose a Multi-armed Bandit (MAB) based data offload

*A. Mukherjee, S. Misra, and V. S. P. Chandra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India.*
*Mohammad S. Obaidat is with the School of Engineering, Nazarbayev University, Kazakhstan.*

path selection scheme, which is governed by the formulated utility function, enabling the selection of resource-optimized paths within the swarm from the source to the target UAV. As per our scenario of communicating data from the UAV in a deployment grid to another UAV in a multi-hop manner, the major challenge is the selection of the UAVs, which will be able to handle the processing under the constraints of limited UAV processing power, limited UAV energy, and path sustainability during data transmission within the edge swarm.

In our scenario, we consider an imaginary aerial plane parallel to the ground, which we divide in the form of an $n \times n$ 2-Dimensional grid as shown in Fig. 1(a). Each grid cell or grid location is assigned a UAV from the swarm. Each of these UAVs is capable of communicating with its immediate one-hop neighbors. The UAV operate in an area, which is divided into grid cells $n \times n$, each of 1 sq.unit area, and these UAVs can communicate directly to its immediate 8 neighbors in the grid layout. Each of these UAVs can act as a gateway between a ground user (restricted to that UAV's grid) and the airborne edge swarm. A user may assign tasks to any of these UAVs, subject to the restriction on the UAVs motion within its grid. In this paper, we strive to choose the best offload path between a source and target UAV (from which the data is requested for the source UAV's grid) without the data leaving the edge swarm or consuming the backbone network's bandwidth.

### 1.1 Offload Path Selection Within the Edge Swarm

The mechanism for offload path selection between a source and a target UAV is outlined in Fig. 1. We consider a $4 \times 4$ grid layout, each with a member of the edge UAV swarm. A source UAV in location $1D$ needs to offload its data to the UAV in grid $4A$ at an instant of time $t_1$. As the communication between the UAVs is restricted to one-hop, the source UAV needs to choose a relay UAV having sufficient energy and processing available to survive the data offload task. Regarding Fig. 1(a), the data from $1D$ is thus offloaded to the UAV in $1C$. This sequence is followed via UAVs in grids $2B$, $3A$, and finally to $4A$, which is the intended target UAV. Similarly, during another task request at a future time $t_n$, a separate path is chosen (refer Fig. 1(b)), which at that instant of time is more resource optimal, with higher chances of survival till completion of the data offloads. It is to be noted that, for this work, we consider that the UAVs can replenish their energy or the swarm can replace the UAV itself upon exhaustion of its energy, within a realistic frame of time. The selection of the UAVs forming the resource-optimized paths is addressed using a utility function tuned MAB path selection scheme discussed in Section 3. The utility-based routing problem in MANETs is functionally quite similar to our approach. The advantage of MAB-based learning is its ability to decide the reward for the formulated path utility. As this is an online learning algorithm, we train the initial stages on a powerful processor (PC or server) and export the trained model to a constrained processor (Raspberry Pi) onboard the UAVs. The models keep on updating as per the changes in the status of the UAVs and the paths formed between them. This approach additionally helps to capture unforeseen changes, which are then used to update the MAB-based model.

### 1.2 Motivation

A typical edge-based UAV in a swarm is highly resource-constrained. Restricting data flow for a local network to conserve backbone network bandwidth and to speed-up transfer times can be achieved by using an edge-based architecture. In such a scenario, task and data offloading within the edge swarm would require high degrees of energy efficient coordination and path planning for routing data packets amongst the swarm members. Established approaches based on shortest path schemes may provide a direct solution to this problem, however at the cost of overburdening only a select few members of the edge swarm, which may lead to a disproportionate and biased distribution of load within the swarm. This biased load distribution proves highly detrimental to the overused UAVs' energy banks and processors. To address this problem, in this work we formulate a multi-armed bandit (MAB) based energy and processing optimal path selection for data offload within an edge swarm of UAVs, which offers proportionate load distribution across the swarm and promises larger mutual savings in energy.

### 1.3 Contribution

In this work, we propose a utility maximization approach for tuning the reward generation of a MAB-based reinforcement learning approach for selecting energy and processing optimized data offload paths between a source and target UAV in a decentralized edge UAV swarm. The following distinct contributions have been made in this work:

- The proposed approach allows for higher energy savings in an ad-hoc edge UAV swarm.
- The proposed approach allows for more uniform task distribution and UAV selection in the network, which allows for reduced wear and tear of the constituent UAVs.
- The selected offload path using our proposed approach ensures the continuity and survival of the chosen path even for larger data sizes.

### 1.4 Paper Organization

The introduction section outlines the problem of selection of collective energy and processing optimized selection of offload path in edge UAV swarms. A literature survey in Section 2 follows this section. Subsequently, Section 3 provides an overview of how our problem can be addressed using MAB. Section 4 outlines the methodology pursued in addressing our problem of offload path selection in edge UAV swarms. Section 5 provides a comprehensive summarization of the results. Finally, we conclude our work in Section 6 with a discussion of the limitations of our approach and future works.

## 2 Related Work

The emergence of mobile IoT systems has resulted in the need for faster, processor-friendly, and energy efficient systems and methodologies for QoS guaranteed communication [2]. One of the upcoming work areas in IoT – data offloading – aims to address these requirements without change of hardware and necessary communication frameworks. Data offloading in mobile edge systems have been used to speed up computation and prolong the battery life of edge devices using time division multiplexing of bandwidth allocation fiber-wireless networks [3]. Various IoT frameworks for data offloading such as femtocell IP access (FIPA) and selective local controller traffic offloading (SLCTO) [4], offloading as a decision problem [5], and the concept of mobility as a service for D2D-based information-centric content distribution [6] have already been proposed.

One of the essential tasks of offloading is the selection of the offload path or the offloading relay/node. Approaches such as the delay and energy-aware data offload relay selection scheme for D2D communication in LTE-A [7], predictive mobility-aware selection of communication paths for migration of virtual machines resulting in a reduction of offload delays and energy consumption [8] present new approaches for addressing the selection problem in data offloading. Various learning-based approaches have been formulated to enhance the robustness of routing solutions. Learning has been used in various routing schemes such as detection and avoidance of sink-hole attacks [9], for enhancing the performance of WMNs by enabling transmission over long paths, island nodes, and interference [10], and improving the networked control of highly mobile unmanned systems [11]. Interestingly, the use of UAVs as aerial base-stations for offloading data at cellular edge networks has also been proposed, the performance evaluation of which has reported significant cost savings and increased network throughput [12]. UAV-based topological path selection approaches in flying ad-hoc networks (FANETs) have been undertaken to make use of Dijkstra's shortest path routing [13].

### 2.1 Synthesis

Most of the works in IoT and edge data offload either deal with a network of static edge devices or standalone mobile edge nodes. Additionally, the use of UAVs (standalone, networked, and swarms), although proven to be quite cost effective and beneficial, are yet to find widespread acceptance in IoT applications. Intelligent and optimized data offload relay or path selection schemes are mainly restricted to ground-based networks. UAV-based path planning, selection, and routing are mainly restricted to physical control of UAV trajectories [1]. In this work, we propose the use of a MAB-based reinforcement learning approach for addressing the problem of energy and processing optimized offload path selection in decentralized edge UAV swarms.

## 3 Multi Armed Bandit Based Offload Path Selection

We divide this section in two parts – 1) *Offload Path Utility Formulation* and 2) *Multi Armed Bandit based Reward Selection*.

### 3.1 Offload Path Utility Formulation

We formulate a single utility function $U_p(t)$ such that it incorporates the factors of residual energy of the UAVs in the edge offload path, the available processing power, the hop distance, and the task load of each UAV processor. We formulate the path utility as:

$$U_p(t) = \begin{cases} \frac{\sum_{i=1}^{n} RE_i(t) * P_i(t)}{n^2 * (1 + \sum_{i=1}^{n} J_i(t))}, & \forall i E_i(t) > T * e_d \\ 0, & \exists i E_i(t) \leq T * e_d \end{cases} \tag{1}$$

where $RE_i(t)$ defines the residual energy of UAV $i$ in the path $p$, $P_j(t)$ defines the processing power available at the end UAV of the path, $n$ is the hop distance, $J_i(t)$ the task list of UAV $i$, $E_i(t)$ the energy present in UAV $i$, $T$ is the time taken for the transfer of data to the target UAV and $e_d$ the energy decrement rate of the UAVs, all at time $t$ in path $p$. As the path utility requirements are directly proportional to the residual energy ($RE(t)$) and processing power ($P(t)$) of the nodes, the product of average residual energies in a path $\sum_{i=1}^{n} RE_i(t)/n$ and average $\sum_{i=1}^{n} P_i(t)/n$ is considered whenever $E_(t) > T * e_d$. The residual energy $RE_i(t)$ is the energy remaining in the UAV $i$ at $t$, whereas the total energy $E_i(t)$ is projected at $t$ based on $T$ and $e_d$. The projected value of $E_i(t)$ provides the estimate to the MAB whether a data transfer will be successful in a chosen path with the existing $RE(t)$ or not (as the time taken to transfer a fixed amount of data is already calculated in Fig. 4(a). The numerator is then divided by the number of tasks/jobs ($J(t)$) to be addressed in that path to estimate the total distribution of tasks to the UAV nodes contained in the path selected. The task list is made up of tasks assigned to the UAVs in the selected path. As the job-list can be zero (if no tasks are assigned to the selected UAVs), the term is incremented by one in the denominator to avoid divide-by-zero errors.

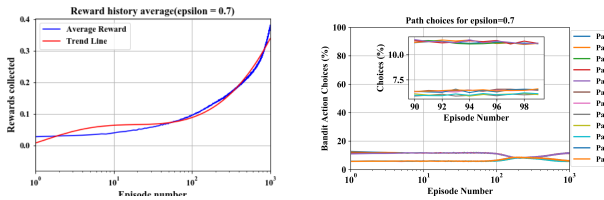
**Lemma 1.** *The collective residual energy $\sum_{i=1}^{n} RE_i(t)$ of a selected offload path decreases with time $t$ such that,*

$$\sum_{i=1}^{n} RE_i(t_1) > \sum_{i=1}^{n} RE_i(t_2) > \sum_{i=1}^{n} RE_i(t_3), \; \forall \, t_1 < t_2 < t_3 \tag{2}$$

### 3.2 Multi Armed Bandit based Reward Generation

We derive our methodology for rewarding the selected path from the classic example of reinforcement learning's Multi-Arm Bandit (MAB) problem, which has three major components – 1) Arms, 2) Agents, and 3) Rewards. The multi-arm bandit consists of $N$ number of arms (bandits), numbered from $1$ to $n, \exists \, n \geq 2$. Each arm has its unknown probability distribution of success $P_{action}$. Selecting the $i^{th}$ arm results in a reward $reward_i$, which is sampled from the distribution of probabilities $P_i$ of the arms. We consider a stochastic multi-armed bandit for this work. An agent has a budget of $k$ arm pulls, which have to be executed in a manner to maximize the accumulated rewards after $k$ arm pulls. Considering the reward for pulling arm $action_i$ at the $t^{th}$ step be $reward_{action_i,t}$, which is sampled from $P_{action_i}$, an agent tries to maximize $\sum_{t=1}^{k} reward_{action_i,t}$.

The non-triviality of the multi-armed bandit problem lies in the fact that the agent cannot access the specific bandit probability distributions as all learning is carried out via the means of trial-and-error and value estimations. In this case, the support of the probability distribution $P_{action}$ is $[0, 1]$, which implies that the probability of success $\mu_{action}$ and the rewards for each arm are bounded between $[0, 1]$. A sub-case of stochastic bandits are Bernoulli bandits, in which the rewards are either 0 or 1 and the probability distribution $P_{action}$ is a Bernoulli distribution with unknown success probability $\mu_{action}$. We utilize the Bernoulli MAB algorithm for our work as shown in Appendix A. As the path selection task in this work is formulated in such a manner that a certain path between a source and target UAV will either be selected or rejected based on the optimizing parameters, a Bernoulli distribution confirms to the need of the problem. The detailed algorithm is outlined in Appendix A.




(a) $Eps = 1000, \epsilon = 0.7, Exp = 100000$ (b) $Eps = 1000, \epsilon = 0.7, Exp = 100000$

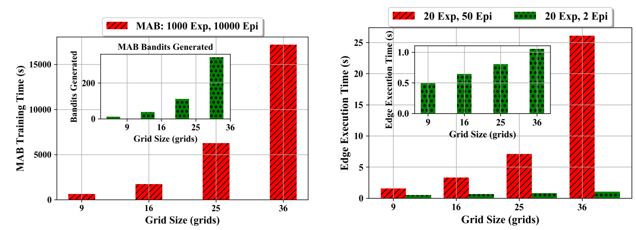
Figure 2: MAB rewards and generated path choices averaged over 1000 episodes and 100000 experiments for $\epsilon = 0.7$ in a grid size of $3 \times 3$.

Our problem statement uses this method of solving the MAB to select the UAV to offload the data for processing. Each **state** of MAB in our problem statement is a data packet waiting to be sent to a target edge UAV node. The state consists of all the information regarding the network architecture, the current parameters of each UAV in the network. The information includes the energy, the amount of data each UAV is already processing and the amount of processing power it can afford. Similarly, each **action** in our problem deals with the selection of the UAV nodes for a path through which the data is to be offloaded between the source and the target UAVs. Each of the available actions is considered while choosing the action for the current state. Finally, we define the **reward** for selecting an offloading path as a utility value which constitutes the information regarding the path between the source and the target UAVs. This information considers residual energies of the UAVs in the path, the amount of processing power available at the UAV that processes the data received, the hop distance between the producer UAV and the target UAV via the path, and the amount of data already in queue to be processed in the UAVs constituting the path. To confirm with the MAB reward, we restrict the utility value into the range of $(0, 1)$ using the sigmoid function as $R_p(t) = e^{U_p(t)}(1 + e^{U_p(t)})^{-1}$. In continuation, we summarize the objective of our work and represent it as:

$$\underset{n,RE_i,J_i,T,e_d,E_i}{\text{Max}} R_p(t) = \frac{exp^{\frac{\sum_{i=1}^{n} RE_i(t)*P_i(t)}{n^2*(1+\sum_{i=1}^{n} J_i(t))}}}{1 + exp^{\frac{\sum_{i=1}^{n} RE_i(t)*P_i(t)}{n^2*(1+\sum_{i=1}^{n} J_i(t))}}} \tag{3}$$

This objective function is subject to the constraints of $E_i(t) > T * e_d,\ \forall i,\ \sum_{i=1}^{n} RE_i(t) > 0\ \forall t \in (0, \infty]\ n > 2,\ \forall\, n \in \mathbb{I}^+,\ J(t) \geq 1, \exists\, J(t) \in \mathbb{I}^+$. As the formulated utility function is strictly convex (refer: Theorem 1), the rewards generated for each path are always maximized, which in turn, allows us to select the most optimized offload path from amongst the choice of multiple offload paths between the source and the target UAVs.

**Theorem 1.** *The formulated path utility function $U_p(t)$ is strictly concave in the interval $(0, \infty]\ \forall\, U_p(t) \in \mathbb{R}^+$.*




(a) MAB Training on Server (b) MAB execution on Edge UAV

Figure 3: Real-life MAB training and execution metrics collected from a server as well as an edge UAV.

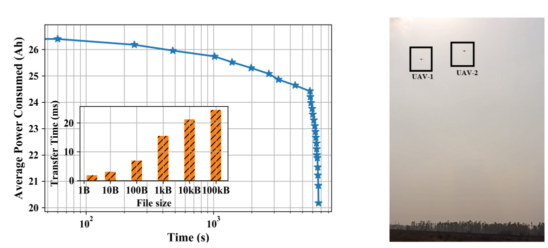
## 4 Methodology



This section outlines the methodology followed in addressing the defined problem using the MAB-based path selection scheme in resource-constrained edge UAV nodes in a UAV swarm. The server-based MAB model training prepares a MAB model to suggest possible paths based on the formulated objective function in Section 3.1. The trained model is exported to the edge UAV systems, enabling a source UAV to utilize the tuned information to select optimal offload path to a target UAV. We obtain various hardware metrics for two networked autonomous UAVs in flight and emulate a large UAV swarm using these metrics.

### 4.1 MAB Model Training

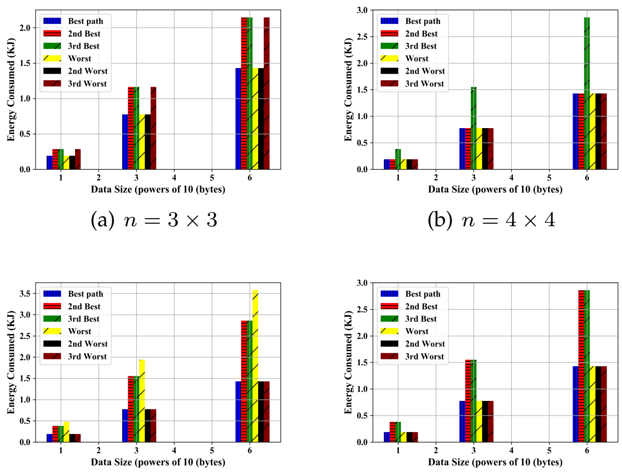
The MAB training is undertaken on an Intel $i3$ quad-core processor with $4GB$ RAM, acting as a networked remote server. Figs. 2(a) and 2(b) show the comparison of the rewards and actions generated (paths selected) averaged over varying experiments, $\epsilon$ and episodes. Fig. 2(a) additionally shows the approximation of the trend line, the average reward follows with each episode. These trend lines are a degree 3 polynomial estimation of the average. We observe that as we increase the number of experiments for the MAB (Appendix D), rapid fluctuations in the generated rewards are reduced, ultimately leading the MAB to follow a smoother reward selection, which follows the average trend line. This trend is also highlighted in the generated




(a) Single Edge UAV performance (b) UAVs in flight

Figure 4: Real-life UAV test-bed implementation.

actions in Fig. 2(b). For the same number of experiments and episodes, we observe a smoother approximation of the rewards for higher values of $\epsilon$ (this trend is also reflected in the generated actions in the form of reduced fluctuations in the bandit choices (shown in the figures in Appendix D). However, on the flip side, increasing the episodes and the number of experiments for each MAB incurs heavy processing and more time is spent on training the model, which makes the trained model unusable for resource-constrained edge devices.




(a) $n = 3 \times 3$ (b) $n = 4 \times 4$

(c) $n = 5 \times 5$ (d) $n = 6 \times 6$

Figure 5: Efficacy of the selected paths in terms of energy consumed for variations in transmitted data sizes analyzed over varying grid sizes for a MAB with 100 episodes and 1000 experiments with $\epsilon = 0.1$.

### 4.2 Small-scale Hardware Evaluation of UAV Links

The MAB model is implemented on an autonomous quadrotor UAV controlled by a Raspberry Pi processor. Each Raspberry Pi hosts the trained model and is in charge of communicating with the other UAVs in its vicinity. We evaluate the power consumption, and data offload metrics between a $3 \times 3$ grid placement of Raspberry Pi units (which are the communicating units in the UAV swarm) in a controlled environment. The power consumption is obtained from live UAV flights and includes the power budget of the UAV as well as the communicating unit onboard each UAV. The hardware metrics for the power consumption and data transfer rate have been averaged out for these nine units in Fig. 4(a). The obtained metrics are used to implement a large-scale UAV swarm emulator, which mimics the behavior of the energy usage profile mirrored in the captured real-life hardware data. The MAB model training time on the server is charted in Fig. 3(a). However, we observe that for increasing grid sizes, the MAB training time rapidly increases, and so do the number of bandits generated. To enable a lighter MAB model on the resource-constrained Raspberry Pi controlled edge UAV node, we choose two models, each trained for 50 episodes over 20 experiments, and 2 episodes over 20 experiments, respectively (shown in Fig. 3(b)). Fig. 4(a) shows the operational battery power consumption profile for the real-life autonomous quadrotor in flight using $12V$, $2200mAh$, $4-$cell Lithium-polymer batteries. For testing, the quadrotors were kept hovering at an altitude of $20m$ above the ground.

### 4.3 Emulation of a Large-scale Edge UAV Swarm

The hardware metrics obtained for data transfer and power consumption between two autonomous UAVs in flight are used for emulating a large-scale edge UAV swarm on a server. We built the emulator on a Python framework, which allows for UAV placement in varying implementation areas with different grid cells. The MAB model is incorporated along with the UAV metrics to study the various effects of parameters such as the MAB model parameters, offload data size, grid size, and UAV replenishment/recharge time. Utilizing the hardware metrics in Fig. 4(a), we implement the data transfer rates for the different data sizes of $1B$, $10B$, $100B$, $1KB$, $10KB$, and $100KB$ in our emulation. Additionally, we also include the battery discharge profile for evaluating the power consumption by the collective network. For normal operations without power replenishment lags, we have considered that a UAV remains offline for 200 seconds, during which its power source is replaced. In continuation, increased duration of power replenishment lag of 2000 seconds is considered in the emulation. Additionally, from actual hardware analysis we found that the processor incurs an average processing overload of $5\%$ of its total processing resources for every connection to it, which has been included in the emulation accordingly. These parameters are analyzed in details in Section 5.

## 5 Results

In this section, we establish the efficacy of the selected paths for use in the data offload scenario. Additionally, in this work, we consider two metrics of the collective residual energy of the network and the number of UAVs alive post completion of the offloading task. Using these metrics, we analyze the performance of the edge UAV swarm by studying the effects of change of offloading data size, increase in the number of grids, and efficiency of the UAV replenishment/ replacement.

### 5.1 Efficacy of the Selected Path

Fig: 5 shows the efficacy of our formulated utility-tuned MAB in selecting the offload paths between a source and target UAV, with the provision of replenishment of UAV or its energy source within an acceptable relative time frame of 200 seconds. Figs. 5(a), 5(b), 5(c), and 5(d) show the collective energy consumed by the UAVs forming the selected offload path for grid sizes of $3 \times 3$, $4 \times 4$, $5 \times 5$, and $6 \times 6$ respectively. We observe that for varying data sizes in all these cases, the first and second choice of paths incurs the least energy consumption during data offload, in turn reinforcing the efficacy of our approach. As this is a one-time data offload process, in some cases, some of the paths (both best and worst) have comparable energy consumption, which is attributed to similar energy profiles at both of these paths resulting in the precedence of additional parameters of availability of processing resource or task list at the UAVs at that instant of time.

### 5.2 Effect of Increase in Offload Data Size

Figs. 6 and 7 the effects of offload data size are evaluated for the total residual energy of the network and number of UAVs alive after completion of data offloading for defined periods of time in a $3 \times 3$ grid space. The proposed approach is compared to the shortest path approach. We observe that for increasing data sizes in a smaller implementation (such as the $3 \times 3$ grid) and smaller operational durations (up to $500s$) MAB performs better than the shortest path approach as seen in Figs. 6(a), 6(b), 6(c), 6(d), 6(e), and 6(f). However, as the operational duration increases, the performance of MAB starts falling for smaller grid sizes. Increasing the grid sizes increases the performance of our MAB-based approach significantly, even for longer operational durations greater than $500s$ (refer to Appendix E).

In continuation, observing the trend of UAVs alive in the swarm in Figs. 7(a), 7(b), 7(c), 7(d), 7(e), and 7(f), we see that the shortest path approach tends to have an almost comparable number of UAVs alive for all the data sizes for all time duration of operation. It is to be noted that the shortest path based approach uses the same set of UAVs for data offload. So, even if a UAV exhausts its energy, it is taken out from the path, and a new shortest path is chosen without considering the requirements of processing and tasks at the new UAVs. After replenishing its energy, the absent UAV returns to the path, resulting in a nearly constant number of UAVs alive in the swarm, even for long durations of offload. In contrast, in our approach, the data offload path continually tends to change due to the changing status of energy, processing and task demands of the constituent UAVs. This approach ensures an even usage of all the UAVs in the swarm without overclocking a select set of UAVs, which in turn, ensures the longevity of the UAVs, unlike the shortest path method. However, for larger data sizes, the shortest path UAVs (with no provision for considering the energy requirements of the UAVs) tend to run out of energy during data offload resulting in better performance of our approach for larger data sizes as highlighted in Fig. 7(f).

### 5.3 Effect of Increase in Grids

Fig. 8 shows the effect of increasing the implementation area or the number of grids, which translates to increasing the number of UAVs in the swarm, with a relative recharge/energy replenishment time of $200s$, with a MAB model trained for 100 episodes, 1000 experiments, and $\epsilon = 0.1$. In Figs. 8(a), 8(c), and 8(e), we see that as the number of grids increase, our approach starts performing significantly better than the shortest path approach, even for longer operational durations for larger data sizes ($> 100kB$). Additionally, the ratio of the number of UAVs alive in our approach to the shortest path approach (refer to Figs. 8(b), 8(d), and 8(f)) increases as compared to the number of UAVs alive in the $3 \times 3$ implementation area. Additionally, for larger implementations ($7 \times 7$, $8 \times 8$, and $9 \times 9$ grids) it is observed that the MAB-based approach

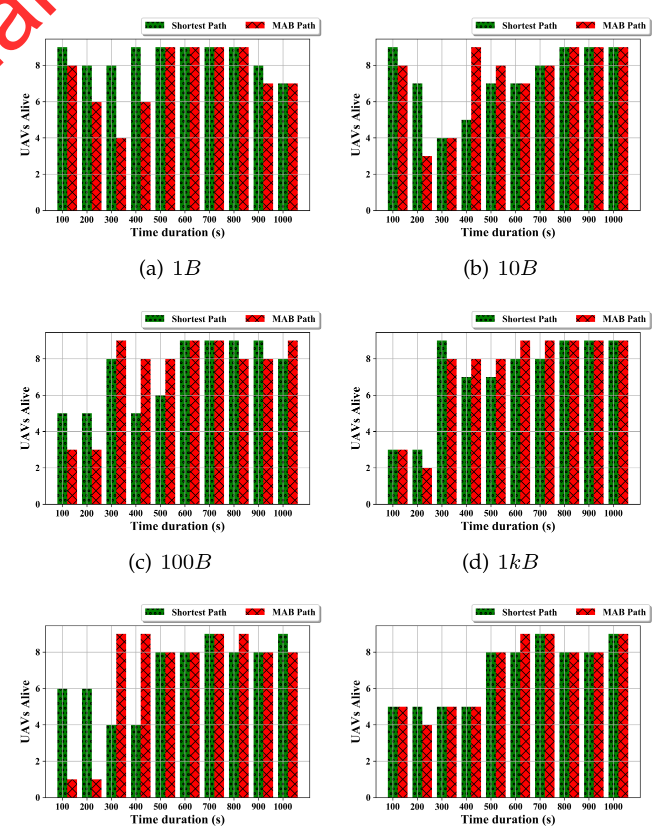



(a) $1B$ (b) $10B$

(c) $100B$ (d) $1kB$

(e) $10kB$ (f) $100kB$

Figure 6: Effect of data size on the $3 \times 3$ edge network residual energy.




(a) $1B$ (b) $10B$

(c) $100B$ (d) $1kB$

(e) $10kB$ (f) $100kB$

Figure 7: Effect of data size on the number of UAVs alive in the $3 \times 3$ edge network.

starts showing significant energy conservation compared to the shortest path approach, and that too from an earlier instant than for smaller grid sizes (refer figures in Appendix E). It is also observed that the difference in the number of UAVs alive for the two approaches become insignificant for larger grid sizes.

### 5.4 Effect of UAV Non-replenishment

Finally, in Fig. 9 we analyse the effect of UAV non-replenishment in the edge swarm for increasing grids using a MAB model trained for 100 episodes, 1000 experiments, and $\epsilon = 0.1$. As there is no recharge/ replenishment involved, the UAVs in the swarm acquire a net negative residual energy balance (refer to Figs. 9(a), 9(c), and 9(e)), except for grid size of $6 \times 6$, where for $100s$ of data offload operation, both the shortest path and our proposed approach have a very minute net positive residual energy balance, as shown in Fig. 9(e).

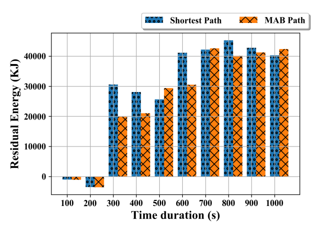
Additionally, we observe that as the number of grids increases, the ratio of the number of UAVs alive in our approach to the shortest path approach significantly increases as shown in Figs. 8(b), 8(d), and 8(f). Comparing the non-replenishment approach in Figs. 9(b), 9(d), and 9(f) one-to-one with the energy replenishment approach in Figs. 8(b), 8(d), and 8(f), we observe that as the grids increase, the performance of our approach keeps improving regarding UAVs alive in the swarm. This result is evident from the significant increase in the ratio of the UAVs alive using our approach and using the shortest approach.

In continuation, summarizing the ratio of UAVs alive (MAB:Shortest Path) for both charging and non-charging approaches during the offload of data sizes greater than $100kB$. In Fig. 10, we observe that for smaller operational times (up to $100s$), the non-replenishment approach has a higher ratio than the approach with replenishment as shown in Fig. 10(a). In contrast, the ratio falls for larger time steps as shown in Fig. 10(b).

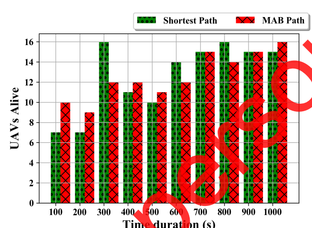
## 6 Conclusion

In this work, we address the resource-optimized data offload path selection between members of a decentralized and resource-constrained edge UAV swarm. The various metrics show the efficacy of our path selection scheme in establishing the most resource-optimized data offload path. Further, large-scale emulation of our approach tuned by real-life hardware metrics collected from multi-rotor UAVs in flight shows the conservation of collective network (swarm) energies, which outperforms approaches such as the simple shortest path based offload path selection. The analysis of the ratio of number of UAVs alive in the swarm for our approach compared against the shortest path approach shows that for larger data sizes and shorter operational durations, our method is comparable to the shortest

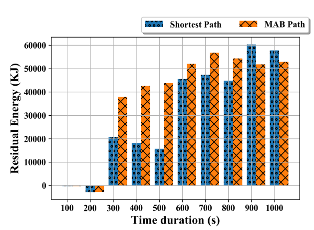




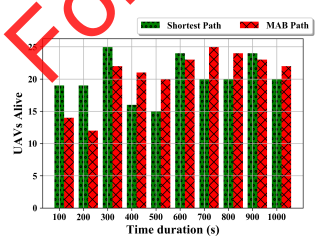
(a) Residual Energy, $n = 4 \times 4$



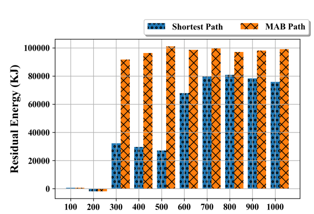
(b) UAVs alive, $n = 4 \times 4$



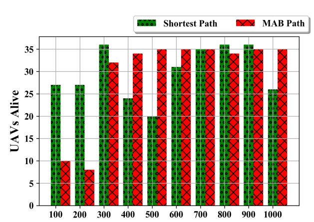
(c) Residual Energy, $n = 5 \times 5$



(d) UAVs alive, $n = 5 \times 5$

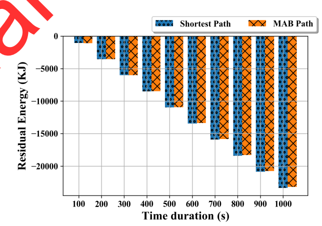


(e) Residual Energy, $n = 6 \times 6$

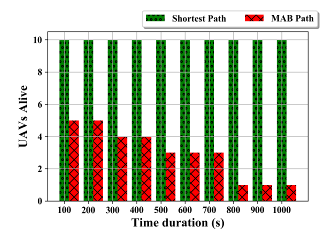


(f) UAVs alive, $n = 6 \times 6$

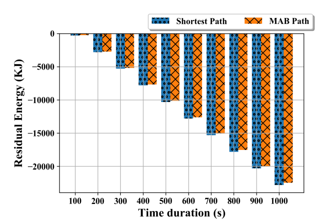
Figure 8: Effect of grid sizes on the edge network residual energy and UAVs alive for a data offload size of $100kB$.



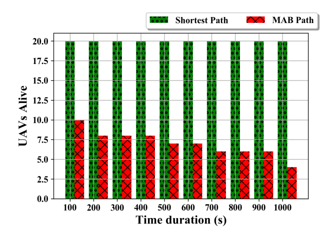
(a) Residual energy, $n = 4 \times 4$



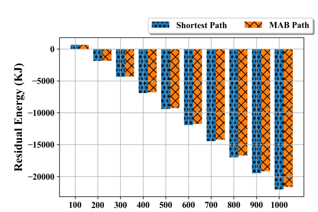
(b) UAVs alive, $n = 4 \times 4$



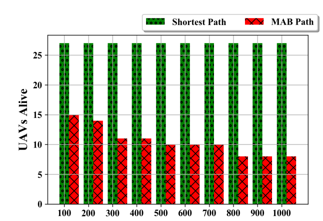
(c) Residual energy, $n = 5 \times 5$



(d) UAVs alive, $n = 5 \times 5$

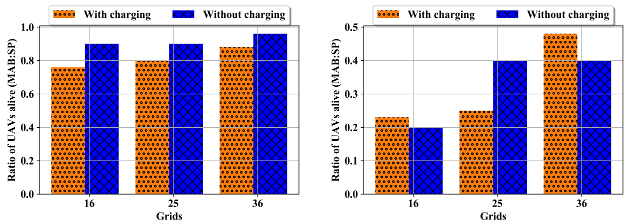


(e) Residual energy, $n = 6 \times 6$



(f) UAVs alive, $n = 6 \times 6$

Figure 9: Effect of large UAV replenishment lag on the edge network residual energy and UAVs alive for a data offload size of $100kB$.




(a) 100 time steps (b) 1000 time steps

Figure 10: Comparison of the UAVs alive ratios of our proposed MAB-based path selection approach to the shortest path selection for UAV replenishment and non-replenishment approaches.

path approach, more so if the provision of UAV energy replenishment is removed. It is noteworthy to mention that as the shortest path approach only accounts for the UAV-UAV hop count, they tend to select the same set of UAVs repeatedly. This approach tends to imbalance the task distribution among UAVs and overworks their electrical systems. This overwork results in the rapid deterioration of the health of a few UAVs, while the remaining UAVs remain unused. In contrast, using the MAB-based approach, the path from the source UAV to the target UAV may not be the shortest, but it is more energy efficient and processor friendly. It is mainly due to this reason that the number of UAVs alive for the shortest path approach tend to be more than the MAB-based approach.

The main limitation of this work is the quasi-static approximation of the edge UAV's mobility. In the future, we plan to incorporate more realistic mobility models with the UAVs and work on offload UAV selection under similar operating conditions of the edge UAV swarm.



## References

[1] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using UAVs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5753–5758.

[2] M. Jutila, "An adaptive edge router enabling Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1061–1069, 2016.

[3] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[4] K. S. Munasinghe, I. Elgendi, A. Jamalipour, and D. Sharma, "Traffic offloading 3-tiered SDN architecture for DenseNets," *IEEE Network*, vol. 31, no. 3, pp. 56–62, 2017.

[5] H. H. Elazhary and S. F. Sabbeh, "The $W^5$ Framework for Computation Offloading in the Internet of Things," *IEEE Access*, vol. 6, pp. 23 883–23 895, 2018.

[6] G. Chandrasekaran, N. Wang, M. Hassanpour, M. Xu, and R. Tafazolli, "Mobility as a service (MAAS): A D2D-based information centric network architecture for edge-controlled content distribution," *IEEE Access*, vol. 6, pp. 2110–2129, 2018.

[7] R. Ma, Y.-J. Chang, H.-H. Chen, and C.-Y. Chiu, "On relay selection schemes for relay-assisted D2D communications in LTE-A systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8303–8314, 2017.

[8] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *2016 IEEE $27^{th}$ Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–6.

[9] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, "Learning for robust routing based on stochastic game in cognitive radio networks," *IEEE Trans. Commun., PP*, vol. 1, 2018.

[10] A. Al-Saadi, R. Setchi, Y. Hicks, and S. M. Allen, "Routing protocol for heterogeneous wireless mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9773–9786, 2016.

[11] W.-S. Jung, J. Yim, and Y.-B. Ko, "QGeo: Q-Learning-Based Geographic Ad Hoc Routing Protocol for Unmanned Robotic Networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2258–2261, 2017.

[12] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Transactions on Wireless Communications*, 2018.

[13] A. Rovira-Sugranes and A. Razi, "Predictive routing for dynamic uav networks," in *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, 2017, pp. 43–47.