UAV Virtualization for Enabling Heterogeneous and Persistent UAV-as-a-Service

Nidhi Pathak, *Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Anandarup Mukherjee, *Student Member, IEEE*, Arijit Roy, *Student Member, IEEE*, and Albert Zomaya, *Fellow, IEEE*

***Abstract*—In this paper, we propose an architecture for UAV virtualization with the primary aim to provide virtualized UAV services to multiple users by envisioning the concept of *UAV-as-a-Service*. In contrast to traditional UAVs, which are resource-constraint in nature and exhibit shorter flight times, our proposed UAV virtualization overcomes the limitations of short flight time of traditional UAVs, in turn allowing them to provide persistent and ubiquitous services. We achieve the virtualization of a UAV through multiple collaborating real-life UAVs performing various tasks in tandem. In this work, we focus on the placement and selection of UAVs to achieve virtualization. We use a social welfare-based approach to select suitable UAVs, from the available ones, and map the UAV to a virtual one. This work enables the provision of different UAV services to multiple end-users, without actual procurement of the UAVs by the end-users. We compare the results for optimal placement, normal maximum energy-based UAV selection, and Atkinson-based selection method. We also compare the virtual model and simple UAV-to-task model to physical UAV usage, task completion ratio, and residual energy of the system. Our proposed model outperforms the traditional model with a task completion efficiency of 94.26%. The residual energy of the system also increases with an increase in the number of tasks.**

***Index Terms*—Unmanned Aerial Vehicle, persistent service, virtualization, scheduling, task allocation, social welfare.**

I. Introduction

THE rapid development of technology in the domain of UAVs, has led to the emergence of crucial application domains such as aerial monitoring, disaster management, search-and-rescue, surveillance, cargo deliveries, and aerial imagery and mapping. However, the limited resources of the UAVs, especially its energy, severely restricts its utility. The low energy capacity and high energy consumption of these UAVs make it challenging to sustain a long UAV-flight duration to complete a task. Typically, a UAV mission is split into smaller sub-tasks and assigned to individual UAVs for completion of the mission, collaboratively [1]. The implementation of automated UAV-battery recharging stations ensures the completion of long duration missions by periodically recharging its batteries [2]. However, such solutions require prior setup of ground control stations, flight operators, and support teams. Due to the significant dependence of these UAVs on the ground control stations, the systems are mostly operated locally within the UAV's communication range. Additionally, these setups are not mobile and dynamic enough to support long-range UAV missions, which restricts the services to only the number of UAVs within the range of the ground setup. In this paper, we propose an architecture to provide persistent and ubiquitous UAV services to the end-users, unlike the traditional UAV services, which are intermittent and short-lived. To enable this proposed architecture, we introduce the concept of UAV virtualization. Fig. 1 depicts the overall system architecture of the proposed UAV virtualization, along with the involved actors. The architecture not only facilitates the services but also provides monetary benefits to the actors involved in this system. Virtualization allows multiple similar UAVs, that may or may not belong to the same owners, to take up a requested task. The proposed architecture consists of three actors– 1) the UAV owners, 2) the service provider, and 3) the end-users. We consider the UAVs are heterogeneous in terms of the number of connected sensors, types of sensor, and battery capacity. The proposed architecture uses the cloud as the backend infrastructure for its implementation. The available

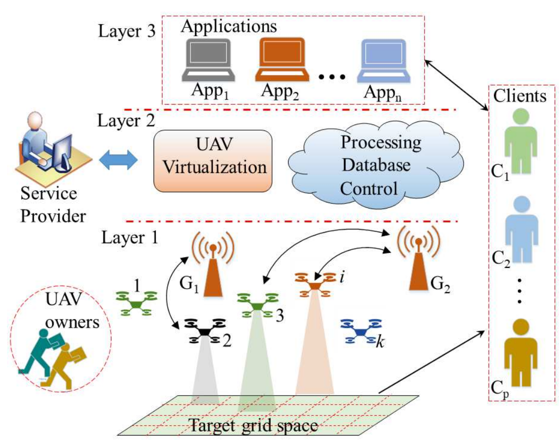


Fig. 1: The proposed system architecture for UAV virtualization.

UAVs which are in the permissible range to take up the tasks form a group, which we term as the local UAV society (soc^i_{local}), specific to that task. These groups are eventually used to select physical UAV and map it to the virtual UAV. A social welfare-based selection scheme is applied to maximize the overall residual energy (E_{res}) of the society. For simplicity, we consider only the energy consumption for UAV traversal and task performance by the sensors.

N. Pathak and A. Roy are with the Advanced Technology Development Center, Indian Institute of Technology Kharagpur, India

S. Misra and A. Mukherjee are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal, India

A. Zomaya is with the School of Information Technologies, The University of Sydney, Australia

A. Motivation

Although UAVs are used to serve a plethora of applications, it is infeasible and uneconomical for an end-user to procure different application-specific UAVs. The operation of multiple UAVs typically requires a team of skilled personnel to deploy, collect, and recharge the UAVs. Additionally, the requirements of prior permission from the government and regulatory bodies before initiating any UAV operations also complicates UAV ownership. The cost of procuring, maintaining, and operating the UAVs are highly prohibitive and cumbersome to manage for a majority of the end-users. The existing shortcomings motivate us to propose the architecture of UAV virtualization to provide seamless UAV services to the end-users. In the proposed architecture, a UAV can serve multiple end-users based on their preferences. The architecture enables an end-user to receive the necessary UAV services without procuring any physical UAVs by them. Additionally, the end-user has a provision to select services among the available ones. On the other hand, UAV owners and service providers receive monetary benefits in place of their services to the end-users. In the traditional UAV service, a UAV owner can serve only a single end-user at a time.

B. Contribution

The proposed architecture provides UAV-as-a-service to end-users through the implementation of UAV virtualization. The specific contributions of this work are listed as follows:

- A generalized UAV virtualization architecture to enable persistent UAV services for long-duration missions and decrease the redundancy in task performance.
- We design an appropriate task-specific UAV selection scheme using social-choice theory, analyze the UAV occupancy and coverage analysis for both homogeneous and heterogeneous UAV types in a defined region.
- We evaluate the performance of the proposed architecture and the selection schemes through rigorous emulation.

II. Related Work

This section highlights the recent developments and research concerning our proposed problem.

A. Sensor Cloud and Virtualization

Sensor cloud has been one of the most demanding areas of research and application since its inception. *Yuriyama* and *Kushida* [3] proposed the idea of making the sensors available and ubiquitous through the sensor cloud infrastructure. Bose *et al.* [4] proposed a scheme for sensor-cloud infrastructure to virtualize the sensors at the application level. Misra *et al.* [5] further proposed refined theoretical modeling of the sensor cloud infrastructure with detailed comparative feature analysis.

B. UAV Cloud

Mahmoud *et al.* [6] proposed integrating UAVs with cloud and incorporated its basic advantages of scalability, high computation resource, high storage, and ubiquity. Further, Luo *et al.* [7] proposed offloading data from the UAVs to the cloud, which releases the onboard memory space for data acquisition. Similarly, other approaches include NFV-based UAV-cloud integration [8] and UAV cloudlets [9].

C. Persistent UAV services

Increasing the capacity of the UAV flight time to produce persistent coverage is one of the most challenging aspects of UAV operation. Lee *et al.* [10] developed a robotic operating system (ROS)-based tracking system, where multiple UAVs can be connected to a central computer and a system responsible for task allocation and handoff among UAVs during a mission, autonomously. Similarly, Park *et al.* [11] implemented a prototype for providing continuous security presence of a UAV, using multiple UAVs in sequence, for a customer in an outdoor scenario.

D. UAV Selection and Task Allocation

Various algorithms are used for UAV selection [12] and task allocation, focusing on different network and flight parameters [13]. Garapati *et al.* [14] proposed reward, penalty, auction-based game theoretic approaches for task allocation models. Kim *et al.* [15] [16] proposed a social choice theory-based selection process focused on the overall resource consumption of a group of robots or UAVs. They used a social welfare function to estimate the overall resource consumption.

Synthesis: The works discussed so far are independently simulated and implemented, with most of them being application specific. Also, the implementation of the works done so far is limited by various factors like ground control station, LoS operation, and autonomous control. A more generic architecture is required with the capability to deal with the issues in a single platform. Towards this aim, we propose a novel scheme in this area targeting the gaps discussed in the preceding sections.

III. System Architecture

The proposed architecture is sub-divided into three layers, as depicted in Fig. 1. Layer 1 constitutes the physical UAVs and the network infrastructure required to connect the UAVs to the cloud server. Layer 2 comprises of the cloud-based operations such as control packets communication, database management, virtual UAV provision, and others. Finally, layer 3 is the application layer, which provides the connecting interface to the end-user and the UAV owners. The functionalities of the three layers are as follows:

Layer 1: A traditional UAV consists of a processor-controller board, rotors, GPS module, and a power source. An additional secondary processor board is attached to the traditional UAV, which connects the sensors and actuators and is used to oversee tasks and perform decision-making during any mission. A communication module is connected to the secondary processor to establish the connection between UAV and the cloud server through the Internet. A physical UAV is equipped with the necessary communication modules that can use both WiFi and cellular network by efficient

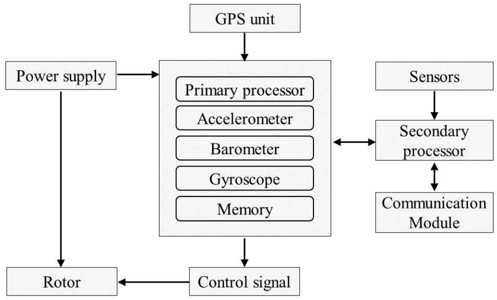


Fig. 2: UAV sub-systems and their interconnections.

switching mechanism to enable its mode of connectivity based on the availability and strength of the network, throughout its operation. Due to highly dynamic nature of the architecture, the data transfer may suffer considerable delay due to the delay in processing, queue, transmission and propagation, given as $\delta_{total} = \delta_{proc} + \delta_{que} + \delta_{trans} + \delta_{prop}$. With continuous connectivity and communication with the cloud, the associated energy consumption also increases, which is the sum of energy required for transmitting data packets from the UAV to the cloud and receiving data packets from cloud to UAV. This energy also includes the periodic update packets with the health status of UAV.

Assumption 1. *A UAV is assumed to be always within the coverage range of a wide-area wireless network (such as WiFi or cellular), which enables the UAV to be connected to the cloud.*

Layer 2: The paper proposes a cloud-based architecture for providing UAV-as-a-service by incorporating the concept of virtualization. The service provider enables a web portal where end-users and UAV owners register themselves. End-users register to receive the UAV services from the service provider. A UAV owner registers to offer their UAV(s) for services to the service provider. A UAV registration includes detailed information about the UAV such as the model, make, battery type, sensors attached to the UAV, the base location of the UAV, also called the home location, and owner. The system assigns the registered UAVs a unique ID at the time of registration. The end-user is required to enter specification and requirement using the templates provided by the service provider. Accordingly, the system generates an end-user-specific application. The cloud server processes the information from the end user's application. The cloud server, through its updated database, checks for the availability of UAV to meet the requirement of the application. The selected physical UAV is then mapped to the virtual UAV, which serves to the application. The complexities of physical UAV monitoring, maintenance, and virtual UAV provisioning is kept abstracted from the end-user. The high mobility and resource constraints of a UAV makes the UAV selection and provisioning more complex. It needs continuous monitoring and database updates. An efficient selection scheme is required to select the most suitable one among the available UAVs while maximizing resource utilization. Layer 2 acts as a middleware layer in the cloud server to enable the process of UAV selection and virtual UAV provisioning. This layer mainly handles the dynamic processes of the architecture and maintains the abstraction.

Layer 3: Layer 3 comprises of the actors involved in the proposed architecture – UAV owners, end-users, and service providers. The end-users can register to the web application with specific details and profile formation, which provides a virtual UAV to the end-user based on the user's requirements. A service provider is responsible for administration and coordination among various components of the architecture, which includes providing a platform for registration of the end-user and UAV owners, database management, UAV maintenance, and virtual UAV provisioning. An end-user or a UAV owner may anytime withdraw their requests and services from the service provider.

We further discuss two different aspects of the virtualization problem. First, we discuss the optimal placements of UAVs in our defined region of interest. Second, the virtualization architecture for a UAV system is defined, supported by a social-welfare-based UAV selection scheme.

IV. Physical UAV Placement

The target region for UAV services is considered to be a $n \times n$ grid space. The placement and operation of the UAVs in a defined grid space are ruled by the conditions and constraints of the architecture. A grid can be populated by only one UAV.

Definition 1. *A n-hop is defined as the distance between a grid and its n^{th} adjacent grid.*

The movement of a UAV is restricted to $1-hop$, where $1-hop$ is the distance between two adjacent grids. Each of the UAVs has a threshold value of energy-level, E_{thr}, below which the UAVs are unable to serve any application.

Let a UAV travel the distance, L_{max}, with a constant velocity v. Also, the maximum energy capacity of the UAV is E_{cap}.

Definition 2. *The per unit energy consumption for the distance traveled by a UAV with its full energy capacity is defined as a metric called energy index, λ_{energy}, of the UAV such that $\lambda_{energy} = E_{cap} L_{max}^{-1}$.*

The total time taken by a UAV during a mission is divided into two parts- t_{travel} be the time required to travel to the target location and t_{hover} be the hovering time utilized for performing the assigned task. The total time required and energy consumed for traversal during a mission by a UAV is given by $T_{total} = t_{hover} + t_{travel}$ and $E_{travel} = v \times T_{total} \times \lambda_{energy}$, respectively. Let, energy consumed for performing a task be E_{task} and the total, E_{tot}, consumed by a UAV during a mission is given by $E_{tot} = E_{travel} + E_{task}$.

Our objective is to minimize the number of UAVs being used for completion of the overall tasks, reduce the overall energy consumption for the tasks generated, and maximize the number of tasks completed subject to certain constraints.

$$min \sum_{i=1}^{Tasks} u_i, max \sum_{i}^{grid} p_{comp}, min \sum E_{travel} + E_{task} \quad (1)$$

TABLE I: Comparison of the proposed architecture with similar technologies.

Parameters	WSN	MWSN	Virtual Sensor Cloud	Traditional UAV	Virtual UAV Service
Field of Operation	Ground, 2D	Ground, 2D	Ground, 2D	Aerial, 3D	Aerial, 3D
Operation Range	Local	Short range	Global	LoS, Short Range	Global
Energy Source	Cell/Li-based Battery	Automotive Battery	Rechargeable cell/Battery	Lipo Battery	Lipo Battery
Lifetime	Limited	Limited	Limited	Limited	Unlimited
Meta data modelling	SML	SML	SML	MAVLink	MAVLink
Mobility	Static	Mobile	Static	Highly mobile	Highly mobile
Deployment Range	Small	Limited	Relatively larger	Limited	Relatively larger
Operating Frequency	2.4/5/	-	-	2.4GHz/5.8GHz	2.4GHz/5.8GHz
Ad-hoc network	✓	-	-	-	-
Hybrid network support	×	✓	✓	✓	✓

subject to

$$1 - hop \leqslant \sqrt{2}L \tag{2}$$

$$x_j.u_{ij} = \begin{cases} u_i, & E_{res}(u_i) > E_{thr} \\ 0, & \text{Otherwise,} \end{cases} \tag{3}$$

$$\sum_{i=1}^{grid} u_i \leqslant 1 \tag{4}$$

UAVs in a group with the same service providing capabilities are termed as homogeneous UAVs while the UAVs with different service providing capabilities are referred to as heterogeneous UAVs. The following subsections discuss the UAV placements of the two categories in an $n \times n$ grid-space where each UAV can travel a distance of $p - hops$.

Definition 3. *A UAV coverage unit is the number of grids that is within its allowed hop distance and can be served for any task.*

A. Homogeneous UAV

First, we define the area covering capacity for a single UAV. Any grid-space can be divided into multiple UAV coverage units.

Theorem 1. *For a p-hop coverage, number of UAVs, U to cover a $n \times n$ grid can be given as* $U = \left(\left\lceil \frac{n}{2p+1} \right\rceil\right)^2$

Proof. For a p-hop coverage, the UAV coverage unit consists of $(2p+1)^2$ grids. An $n \times n$ grid space can be seen as a combination of multiple UAV coverage units represented by a linear combination as:

$$n = (2p+1)k + m, \quad k \in \mathbb{R}_+ \quad and \quad m \in \mod (2p+1) \tag{5}$$

The UAV coverage unit-based grid space representation the minimum number of UAVs required to cover the grid space. We use the ceil value of $2p+1$ to include the case where n is not a multiple of $2p+1$. Let $p = 1$, then $U = \left(\left\lceil \frac{n}{3} \right\rceil\right)^2$. □

B. Heterogeneous UAV

In the case of heterogeneous UAVs, there is a trade-off between the number of UAVs and types of UAVs that can be placed in a grid space. However, the idea is to provide each type of service to all the grids in the grid space.

Theorem 2. *For a p-hop UAV, maximum number of heterogeneous UAVs that can be placed in a $n \times n$ grid space with no wastage of coverage area is* $(n - (2p+1) + 1)^2$.

Proof. A UAV at the boundary of a grid space or a distance less than its hop distance from the boundary wastes some of its coverage grid. We find all the grids that are at a minimum distance of p-hop from the boundary of the grid space. This quantity can be expressed in terms of the hop distance p and grid size n as $U_{max} = (n - (2p+1) + 1)^2$. □

Lemma 1. *Number of under-utilized grids if all the grids are covered is given by* $W = (n + 2p)^2 - n^2$.

Proof. The UAVs at the edges of the grid space cover only the area inside the grid space while the other half is under-utilized, i.e., the grids are under the coverage of a UAV but are not being served. □

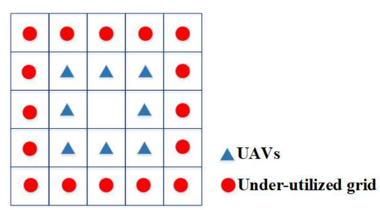


Fig. 3: Under-utilized UAV coverage for one-hop UAV.

Lemma 2. *Maximum number of type of UAVs in a $n \times n$ grid space, if no same type of UAV are allowed to overlap their coverage is given as* $(2p+1)^2$, $\forall\ n \geq (2p+1)$.

Proof. Considering the smallest unit, i.e., the UAV coverage unit, the maximum number of heterogeneous UAVs that are placed without any similar overlapping grids is the total number of grids in the coverage unit. For a 3×3 grid space, the total number of different types of UAV is nine since there are nine grids in the unit. As discussed earlier, any grid space can be represented as a combination of multiple UAV coverage units. Therefore, the maximum number of heterogeneous UAVs for any grid space is always equal to the number of grids in its UAV coverage unit. □

V. Virtualization

A UAV owner is denoted as $o_i \in O$, where O is a set of all the UAV owners. UAV owners lease their respective UAVs to the service providing platform. These UAVs are used in the composition of the virtual UAV for the end-users. Each UAV, registered with the UAV service providing platform is assigned a unique $uid_i \in UID$. A UAV contains multiple homogeneous or heterogeneous sensors. Set of sensor types is defined as $ST = \{s^t_1, s^t_2, s^t_3 \dots, s^t_n\}$. A set of sensor is defined as $S = \{s_1, s_2, s_3 \dots\ s_k\}$ where each sensor s is a 3-tuple represented as $s = \langle id, s_t, \mathcal{S}\rangle, s_t \in ST$. The tuple- id is a unique identifier allocated by the system to the sensor. Any location is represented as a 2-tuple $loc = \langle lat, lon\rangle$, where the lat and lon represent the latitude and longitude values of the location, respectively. The availability of a UAV for certain application depends on its state, $\mathcal{U}$ such that

$$\mathcal{U} = \begin{cases} -1, & \text{in flight} \\ 0, & \text{unavailable} \\ 1, & \text{available} \end{cases} \tag{6}$$

Based on the different UAV-related attributes, a UAV and an application are defined as:

Definition 4. *A UAV is defined as a 5 tuple and represented as* $uav_i = \langle uid, S_i, Loc, E_{res}, \mathcal{U}_i, Loc_h\rangle$*, where* Loc_h *is the home location of the UAV. A UAV can have multiple sensors, represented as a set* $S_{uav} = s_i$*,* $S_{uav} \subset S$*. Similarly, an application is represented as a 3-tuple* $App = \langle A_{id}, A_{type}, A_{loc}\rangle$*.*

A_{id} is the application ID provided by the system at the time of registration, A_{type} is the type of application depending on the task requested by the application. It may be any kind of sensing or actuation, capturing image, atmospheric sensing, mainly related to the type of sensors to be used for the application. A_{loc} is the location of the task requested by the end-user. The set of physical UAVs and virtual UAVs are denoted as $pUAV$ and $vUAV$, respectively.

A. UAV Selection and Allocation

The inputs from the end-user in their application along with the system data about the available UAVs are used for the selection and allocation of UAVs to an application. Fig.

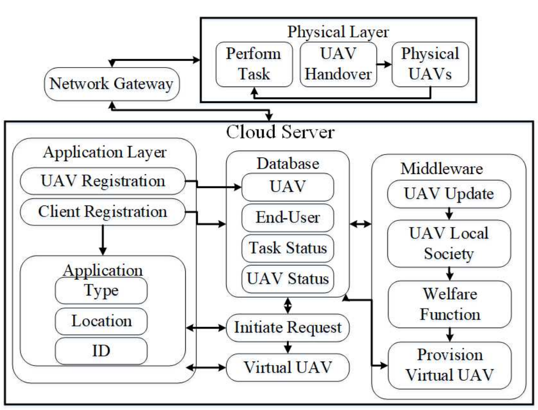


Fig. 4: UAV selection and task completion process.

4 represents the flow for the selection process at different components level. First, the end-user initiates a request for a UAV service through its application. A physical UAV is selected and mapped it to the virtual UAV for an application. The system uses the input from the end-user and determines the type of sensors required for the application. The system goes through several transitions, where each step ensures that the most suitable UAV is selected for the task. The selection is based on certain parameters and index along with the selection algorithms.

We use a function f for selecting suitable sensor nodes for the application, requested by the end-user. Function f is represented in EquationV-A, where A_{type} is the application type ST_{app} is a subset of ST,

$$f(A_{type}) = \{ST_j \mid ST_j \in ST = ST_{app}\} \tag{7}$$

After carefully selecting the type of sensors for an application, a set of UAVs is selected from the available UAVs with the service provider. The function g_1 finds all the UAVs with the type of sensors required and the home location in a range of the location of the application. It also checks for the state of the UAV to avoid selecting the unavailable UAVs.

$$\begin{aligned} g_1(ST_{app}, A_{span}) = \{&uav_i \mid uav_i.s.st \subset ST_{app}, uav_i \in UAV \\ &uav.loc \in range(A_{loc}), uav.us \neq -1\} \\ &= UAV_{app} \end{aligned} \tag{8}$$

The output of function g_1 is a set of UAVs, UAV_{app}, which is the set of all possible UAVs that are eligible for use in the application. The local UAV society is further used in the social-choice based selection algorithm.

B. Social Welfare Function

The social-welfare function, borrowed from the social-choice theory of Economics, is widely used in different fields of application. In a social welfare function, a group of agents votes for their preferable options, among the available ones. The aim of this function is to make the voting and selection process unbiased and equally distributed, towards the holistic welfare of the society. There are different models and functions available to calculate the welfare parameter of a society based on the affecting parameters and the required output. We use the Atkinson index-based social welfare function [17] to analyze the resource utilization of the UAVs. Atkinson index based welfare function has been used for multi-robot task allocation problems [15] [16]. Atkinson welfare function model offers the flexibility to vary the magnitude of the penalty for maintaining equal resource utilization in a society. Based on the Atkinson index [16], we derive our welfare function as:

$$w^s = \frac{1}{n_u} \sum (r^u_i)^{(} 1 - a_k) \tag{9}$$

where n_u, r^u and, a_k are the number of UAVs in the local society, resource value for each UAV in the society and the Atkinson inequality aversion parameter, respectively. In general, the values of aversion parameter used are 1, 1.5, 2, 2.5. When a_k=1, the welfare function is represented as:

$$w^s = exp\left(\frac{1}{n_u}\sum(r_i^u)\right) \quad (10)$$

The eligible set of UAVs is further refined by selecting a UAV in any other application with similar task assignment and location. If any such UAV is found, it can be assigned to multiple applications. Finally, the allocation function f_{alloc} selects the UAV based on the social welfare function and allocates it to the application.

$$\begin{aligned} f_{alloc}(App) &= f(g_1((UAV_{app})) \\ &= \{uav_i \mid uav_i \in UAV_{app}, uav_i, \\ E_{res} > \ E_{hop} + E_{th}, d(uav_i, A_{loc}) &= d_{min}\} \\ &= UAV_{vir} \end{aligned} \quad (11)$$

The eligible set of UAVs and the finally selected UAV are recorded in the application for any future requests. The application is updated periodically or on-demand whichever is best suited depending on the frequency of application usage.

During each cycle of request triggered by the end-user through the application, a UAV from the set of physical UAVs is allocated to the application and mapped to the virtual UAV created for that application, UAV $\rightarrow$ vUAV.

$$f_{virtual}(UAV_{vir}) = \{vuav_{app} \mid UAV \rightarrow vUAV\} \quad (12)$$

In case of long duration mission, the mapping of virtual UAV is repeated when a UAV is worn off its energy and needs another UAV to take over. This is done without notifying the end-user.

As depicted in Fig.4, the selection process is followed by the completion of a task, requested by an application. After the successful mapping of physical to virtual UAV, the UAV flies to the target location and performs the task until it is complete. The platform keeps a check on the energy level of the UAV to prevent the UAV from depleting its battery below the threshold level. When single UAV is unable to complete the task, it can be replaced by another physical UAV. The handover of UAVs is managed by the UAV service platform, autonomously, by analyzing the data from the UAV and selecting a replacement UAV, without any discontinuity in the service.

VI. Performance Evaluation

A. Simulation

We simulate our architecture in a python environment. We generate tasks in an application area split into $n \times n$ grid-space.

The initial energy of each UAV is 10 units. The randomly generated tasks are assigned an energy consumption value for each type of sensor associated with the task. We vary the number of UAVs and applications till 1000, in a grid-space of 50×50 to 500×500, with an Atkinson index of 1.25 to 3.00.

For each task, a local society of UAVs is generated, called UAV cluster. Further, the UAV selection algorithm 1 uses the Atkinson index based welfare function to calculate the overall welfare value for each UAV in the UAV cluster. The UAV with maximum welfare value is selected for the task. We evaluate the performance of our architecture by varying the parameters- grid size, number of tasks, number of UAVs, and the aversion parameter.

Algorithm 1 UAV Selection

```
INPUT: n : Specified n × n grid space, p : Number of tasks, u : Number of UAVs ,
OUTPUT: Tasks completed by UAVs
for Each task do
    Find the local UAV society
end for
for Each local UAV society do
    if Local UAV society is not empty then
        for For each UAV in local UAV society do
            main UAV = UAV
            Calculate the residual energy of the society
            if e_res > e_max then
                e_max = e_res
            end if
            Task completed
        end for
    end if
end for
```

Algorithm 2 Virtualization

```
INPUT: n : Specified n × n grid space, p : Number of tasks, u : Number of UAVs,
OUTPUT: Tasks completed by Virtual UAVs
for i = 1 to p do
    Assign virtual UAV to each task
end for
for Each task do
    Find the local UAV society
    Assign list of UAV society to Virtual UAVs
end for
for For each task do
    if Task location not served already then
        Find physical UAV and map to virtual UAV
        Check energy
        while Task not complete do
            Find physical UAV and map
            Check energy level and perform remaining task
        end while
    end if
end for
```

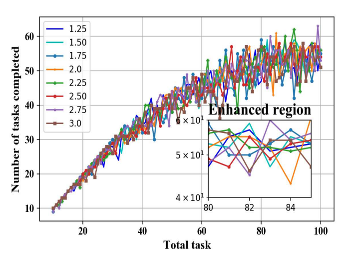


Fig. 5: Tasks completed for varying aversion parameter.

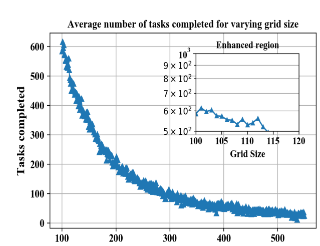
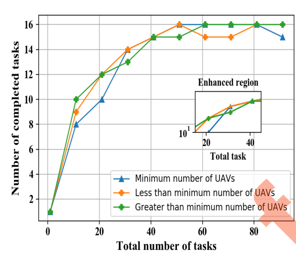


Fig. 6: Tasks completed for varying grid size.

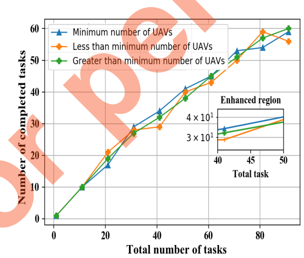
B. Result

We evaluate the performance of the selection algorithm for the four parameters- number of UAVs, number of tasks, size of the grid-space, and the Atkinson aversion parameter (a_k). For different values of the aversion parameter, the penalty imposed upon the society for equal distribution of resource increases. From the plots in Fig. 5, we observe that an aversion value between 1.75 and 2.25 yields the maximum number of completed tasks for our architecture with virtual UAVs. Fig. 7 shows the task completion rate against a varying number of UAVs with a 50×50 grid-space and 1000 tasks, with different values of a_k. As the number of UAVs increases in the grid-space, the size of the local UAV society for each task increases. The number of suitable UAVs for the task increases the task completion ratio. While one UAV per grid assures that all the tasks are completed, the simulation results reflect that a number less than the grid space can be chosen such that a maximum number of tasks are completed as in Fig. 7.

Next, we evaluate the number of tasks completed for a fixed number of 1000 UAVs, 1000 tasks, and varying grid-sizes. As the grid-size increases, the UAV and task distribution become more sparse across the grid space. The size of the local UAV society decreases, often resulting in no UAV available for a task. Therefore, we see a gradual decrease in the number of completed tasks in Fig.6. For fixed grid size and number of UAVs, the number of tasks is varied in Fig. 8. As the density of task increases in the grid space, more tasks are available within the range of UAVs, which increases the probability of a UAV being part of local society. As a result, the number of completed tasks increases. However, it must be noted that the overall task completion ratio decreases with increasing task density over a region. Fig. 2 follows a linear trend for task completion as the number of tasks and grid-size is kept fixed with varying number of UAVs. We attribute this trend to the increasing number of eligible UAVs and size of the local UAV society. A large number of UAVs will contribute to larger residual energy of the local UAV society, resulting in more number of tasks which are completed.



(a) Task completed for varying number of UAVs without Virtualization



(b) Task completed for varying number of UAVs with Virtualization

Fig. 7: Tasks completed for varying number of UAVs.

We compare the performance of simple UAV-to-task completion architecture with the proposed virtualization architecture. In the simple UAV-to-task architecture we assume that only a single UAV can complete a task, i.e., a UAV with energy greater than or equal to the required energy can take up a task without distributing it to multiple UAVs. For Virtual UAVs, the physical UAVs are selected and assigned to the task based on Equation (11). As the number of task increases, multiple tasks get generated on the same grid location. In virtualization architecture, the task at a grid is performed only

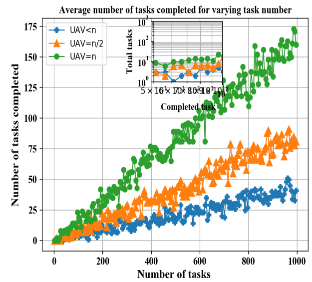


Fig. 8: Tasks completed for varying task count.

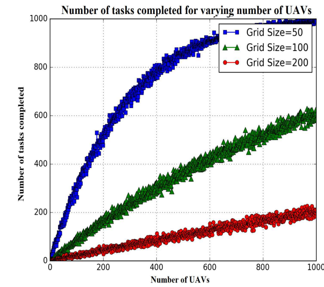
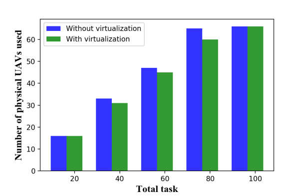
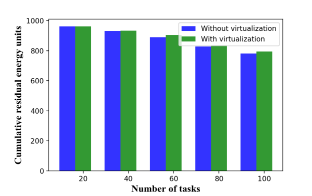


Fig. 9: Task completed for varying UAV count.

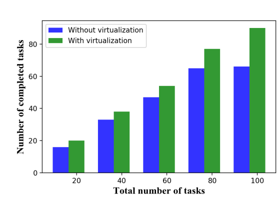
once at the physical level such that the same physical UAV maps to virtual UAVs of all the tasks at the target location. This decreases the number of physical UAVs in use for the tasks, which we observe in Fig. 10a. As a result, the overall residual energy of the system increases, as evident in Fig. 10b. Virtualization also allows the UAVs to take up a task without having full energy to complete it single-handedly, which is not possible in the simple UAV-to-task architecture, leading to more number of completed tasks. Fig. 10c compares the number of tasks completed for the two architectures. The essence of the proposed architecture holds for scenarios with multiple tasks in a grid location. For a condition with one task per grid location, the number of physical UAVs for both the architectures is similar to each task has to be performed individually for both the architectures. We speculate this as the cause for the behavior of our plots in Fig. 10a and Fig. 10b, where values are same for both the architectures with task count of 20 and 100. However, the number of tasks completed in virtualization architecture surpasses the same in the other architecture because of its ability to perform a partial task by a UAV.



(a) Number of physical UAVs used



(b) Residual energy of the system



(c) Tasks completed

Fig. 10: Comparison of virtual architecture with simple UAV-to-task architecture.

VII. Conclusion

In this paper, we proposed a novel scheme for persistent and ubiquitous UAV services through virtualization of physical UAVs. Our work enables the provision of UAV-as-a-service, overcoming the barrier of short flight time of a single UAV. Virtualization also increases the utilization of resources, avoiding multiple UAVs for redundant services. We incorporate a social welfare-based selection process for UAV selection. In the future, we plan to extend the work considering heterogeneous types of UAVs with different sensors onboard. This will challenge the selection algorithms and find the best possible UAVs within the target range. It is important to note that UAV operations with virtualization may raise other concerns and issues that need to be addressed. Service provisioning related parameters can be evaluated, and more robust schemes can be introduced.

References

[1] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim, "Persistent UAV Service: An Improved Scheduling Formulation and Prototypes of System Components," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 221–232, Apr 2014.

[2] A. Trotta, M. D. Felice, K. R. Chowdhury, and L. Bononi, "Fly and recharge: Achieving persistent coverage using Small Unmanned Aerial Vehicles (SUAVs)," in *proceedings of the IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[3] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing," in *proceedings of the 13^{th} International Conference on Network-Based Information Systems*, Sept 2010, pp. 1–8.

[4] S. Bose and N. Mukherjee, "Sensiaas: A sensor-cloud infrastructure with sensor virtualization," in *proceedings of the 3^{rd} International Conference on Cyber Security and Cloud Computing (CSCloud)*, June 2016, pp. 232–239.

[5] S. Misra, S. Chatterjee, and M. S. Obaidat, "On Theoretical Modeling of Sensor Cloud: A Paradigm Shift From Wireless Sensor Network," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1084–1093, June 2017.

[6] S. Y. M. Mahmoud and N. Mohamed, "Toward a Cloud Platform for UAV Resources and Services," in *4^{th} IEEE Symposium on Network Cloud Computing and Applications (NCCA)*, June 2015, pp. 23–30.

[7] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A UAV-Cloud System for Disaster Sensing Applications," in *81^{st} IEEE Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.

[8] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A nfv system to support configurable and automated multi-uav service deployments," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ser. DroNet'18. ACM, 2018, pp. 39–44.

[9] S. Jeong, O. Simeone, and J. Kang, "Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, March 2018.

[10] B. H. Y. Lee, J. R. Morrison, and R. Sharma, "Multi-UAV control testbed for persistent UAV presence: ROS GPS waypoint tracking package and centralized task allocation capability," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 1742–1750.

[11] H. Park, B. H. Y. Lee, and J. R. Morrison, "Persistent UAV security presence service: Architecture and prototype implementation," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 1800–1807.

[12] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-Efficient Computation Offloading for Secure UAV-Edge-Computing Systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.

[13] S. Park, C. S. Shin, D. Jeong, and H. Lee, "DroneNetX: Network Reconstruction Through Connectivity Probing and Relay Deployment by Multiple UAVs in Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, Nov 2018.

[14] K. Garapati, J. Roldán, M. Garzón, J. del Cerro, and A. Barrientos, "A Game of Drones: Game Theoretic Approaches for Multi-robot Task Allocation in Security Missions," in *ROBOT 2017: Third Iberian Robotics Conference*, A. Ollero, A. Sanfeliu, L. Montano, N. Lau, and C. Cardeira, Eds. Cham: Springer International Publishing, 2018, pp. 855–866.

[15] M. H. Kim, S. P. Kim, and S. Lee, "Social-welfare based task allocation for multi-robot systems with resource constraints," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 994 – 1002, 2012.

[16] M. Kim, H. Baik, and S. Lee, "Resource Welfare Based Task Allocation for UAV Team with Resource Constraints," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3, pp. 611–627, Mar 2015.

[17] A. B. Atkinson, "On the measurement of inequality," *Journal of Economic Theory*, vol. 2, no. 3, pp. 244 – 263, 1970.